

AIL Framework for Analysis of Information Leaks

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



CIRCL

Computer Incident
Response Center
Luxembourg

Alexandre Dulaunoy

alexandre.dulaunoy@circl.lu

Aurelien Thirion

aurelien.thirion@circl.lu

info@circl.lu

February 3, 2025

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
- Online chat <https://gitter.im/ail-project/community>



Legal and Ethics

Privacy, AIL and GDPR (PII)

- 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 regulation¹.

¹<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**².

²<https://www.acm.org/code-of-ethics>

<https://www.first.org/global/sigs/ethics/ethics-first>

Introduction

Concepts - Deep Web

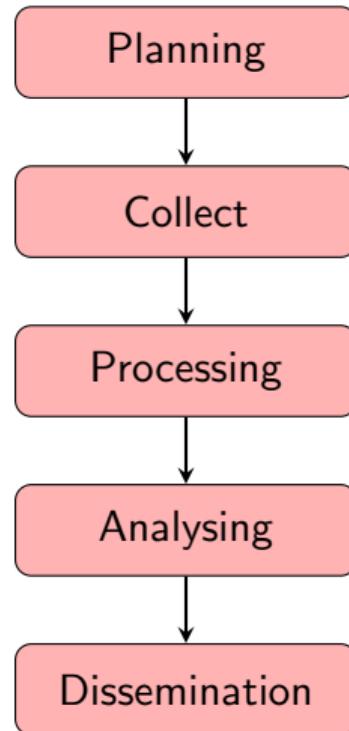
- **Deep Web** is the part of World Wide Web not indexed or directly accessible by standard web search-engines;
- This can be content hidden from **crawlers** by requiring a specific access and this can includes private social media, password-protected forums or content protected by different measures such as paywalls or specific security interface to access the information;
- A large portion of content accessible via Internet is part of the deep web³.

³also called invisible web, hidden web or non-indexed web

Concepts - darknet

- **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



Collecting, processing and analysing content - web pages

- 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 project⁴**.
- In 2023, AIL framework version 5.0 released with new datastorage back-end.

⁴<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" entered in the search bar. The results page displays 322,302 commit results. Three specific commits are highlighted:

- Make remove_password actually work**
Committed by javitonino on March 1. The commit hash is def411c.
- remove password**
Committed by wenlei on March 1. The commit hash is e9611e0.
- remove password**
Committed by yejune on March 2. The commit hash is 037b956.

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 paste service interface. It displays two separate code snippets as "text" items.

Left Item (text, 4.41 KB):

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

Right Item (text, 2.02 KB):

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

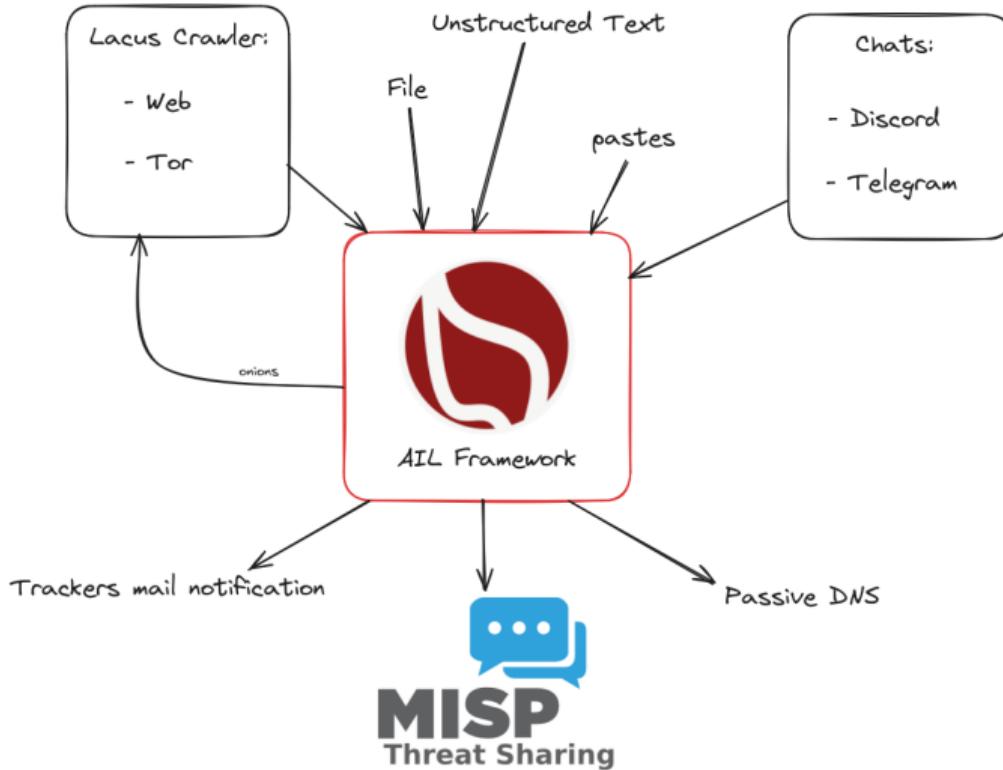
As a CSIRT, we must address this issue.

- **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 booters⁵ or the resale of personal information - comparing reality versus media coverage)
- **Analyzing collateral effects** of malware, software vulnerabilities, or data exfiltration

⁵<https://github.com/D4-project/>

Current capabilities

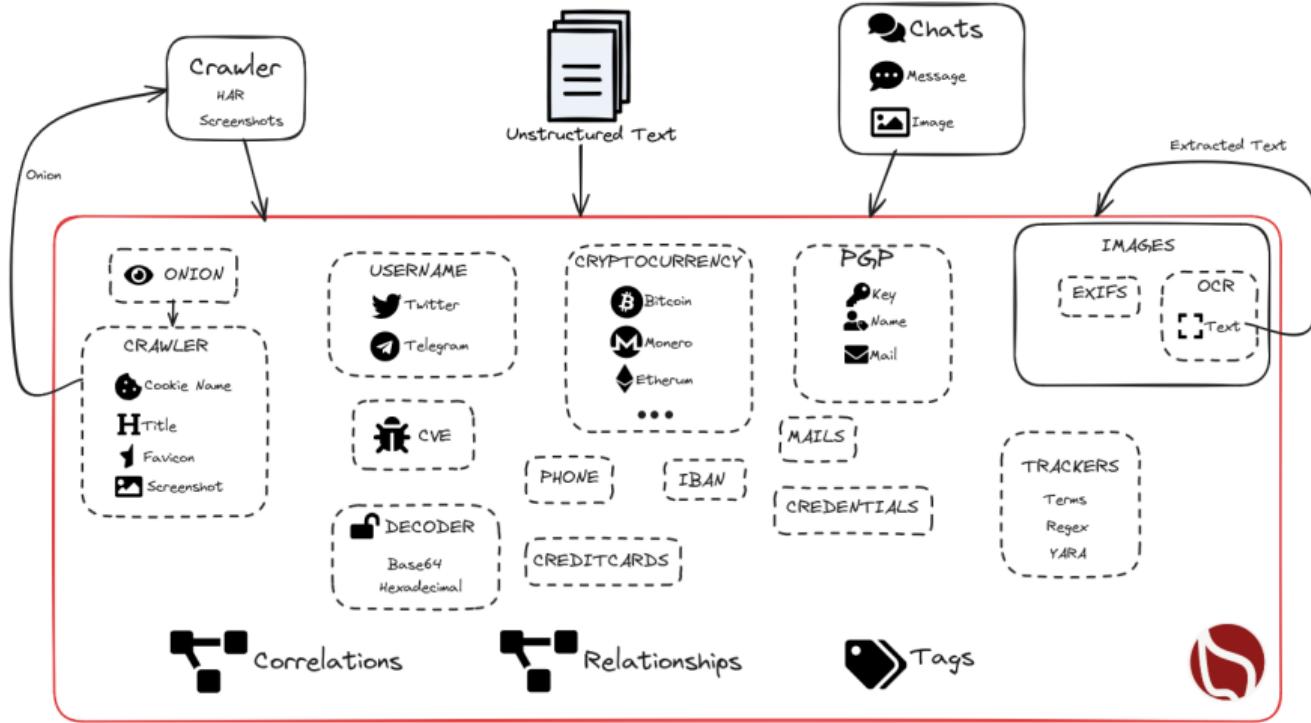
High Level Overview



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

Analysis of unstructured information



AIL Framework: Features

- 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

Trackers - Retro Hunt

- 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: [ail-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** **item**

[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
        $dss_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](#)



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 Sources to track (ALL IF EMPTY)

PGP

Filter PGP by subtype:

name

mail

Tags

Custom Tags (optional, space separated)

Selected Tags

Taxonomic Selected

Selected Tags

Galaxy Selected

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
        Sopenssh_priv = "BEGIN OPENSSH PRIVATE KEY" wide ascii nocase
        Sdsa_priv = "BEGIN DSA PRIVATE KEY" wide ascii nocase
        Sec_priv = "BEGIN EC PRIVATE KEY" wide ascii nocase
        Spgp_priv = "BEGIN PGP PRIVATE KEY" wide ascii nocase
        Spem_cert = "BEGIN CERTIFICATE" wide ascii nocase
        Spkcs7 = "BEGIN PKCS7"

    condition:
        any of them
}
```

Type	Id	Tags
●	archive/gist.github.com/2023/04/14/fuizmiranda7_3b3d1133a3d3842092c5fc5fb39e84f2.gz	infoleak:automatic-detection="private-key" test23 test12 infoleak:automatic-detection="certificate"
●	submitted/2023/04/20/submitted_cc9190ab-80d2-4d2b-9c9e-97c51e69a855.gz	infoleak:submitted="true" test12 infoleak:automatic-detection="rsa-private-key" infoleak:automatic-detection="ssh-private-key" test23 infoleak:automatic-detection="certificate" infoleak:automatic-detection="onion"
●	archive/gist.github.com/2023/04/13/chipzoller_d8d6d2d737d02ad4fe9d30a897170761.gz	test12 test23 infoleak:automatic-detection="certificate"

Recon and intelligence gathering tools

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

Recon and intelligence gathering tools

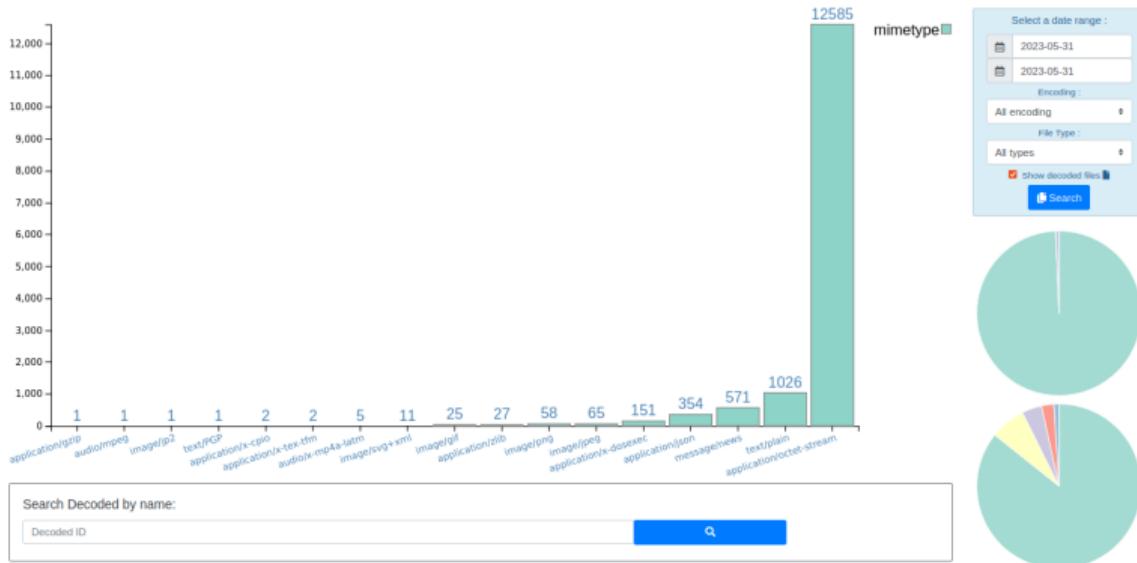
```
#####
=====
Hostname      www.pabloquintanilla.cl           ISP      Wix.com Ltd.
Continent     North America          Flag
US
Country       United States        Country Code   US
Region        Unknown            Local time    19 Nov 2019 07:59 CST
City          Unknown            Postal Code   Unknown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

Decoder

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

Decoder:



20230531 Decoded files:

Show 10 entries

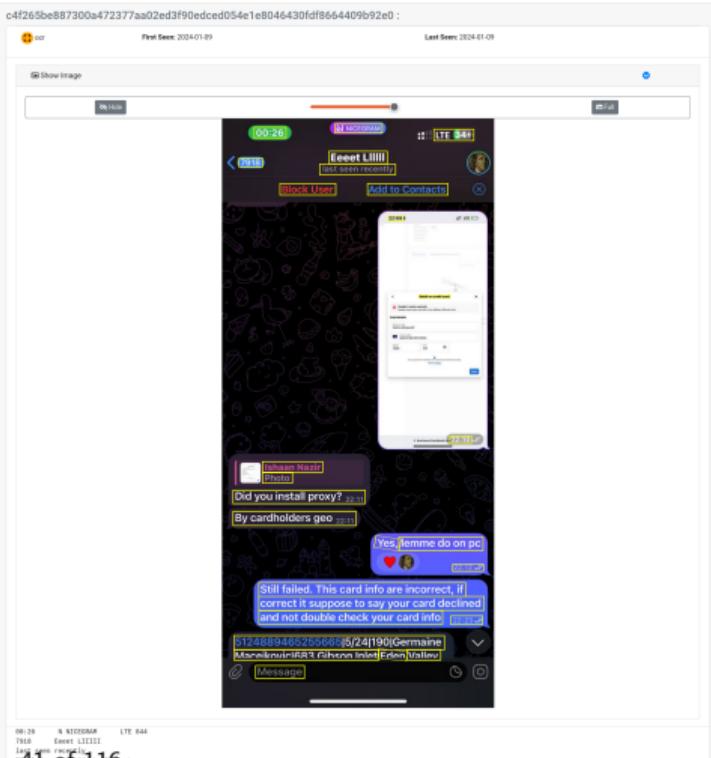
Search:

estimated type	hash	first seen	last seen	nb item	size	Virus Total	Sparkline
image/gif	ee0bc07fe0web22c0ec1364e4bf2e840dc3e05	20230404	20230531	214708	1108	Virus Total submission is disabled	
image/png	b009399cefa0e82086453da04a887105ca26e4	20230404	20230531	8404	1054	Virus Total submission is disabled	
application/json	9fb9180x5b0a395e2523bt44ec6bd2c0k11f9	20230410	20230531	3947	44	Virus Total submission is disabled	

Decoder:

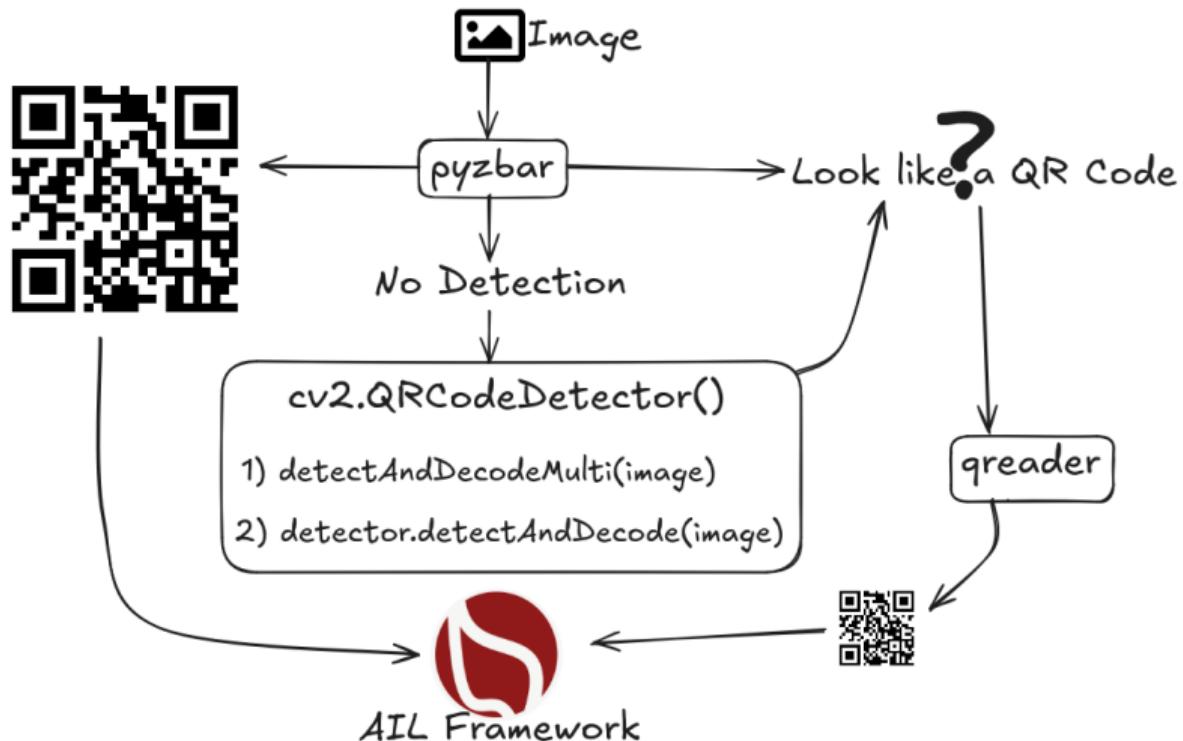


OCR: Optical Character Recognition



- 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.**

QR Code Extractor



QR Codes



2022-12-16
16:51:04

blue_forever20



QR

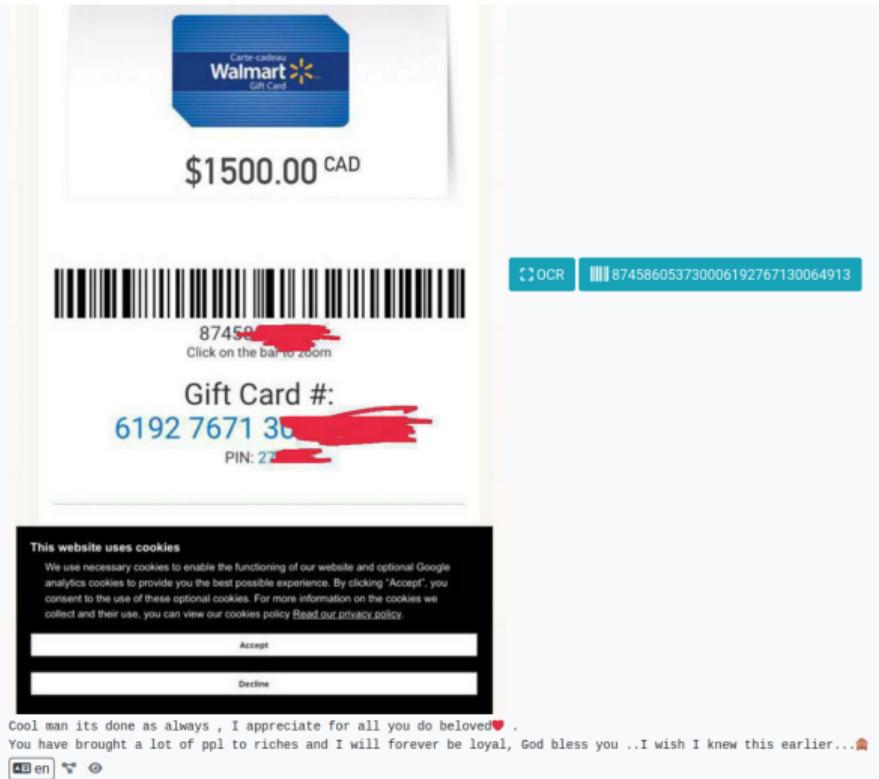
likeaboss



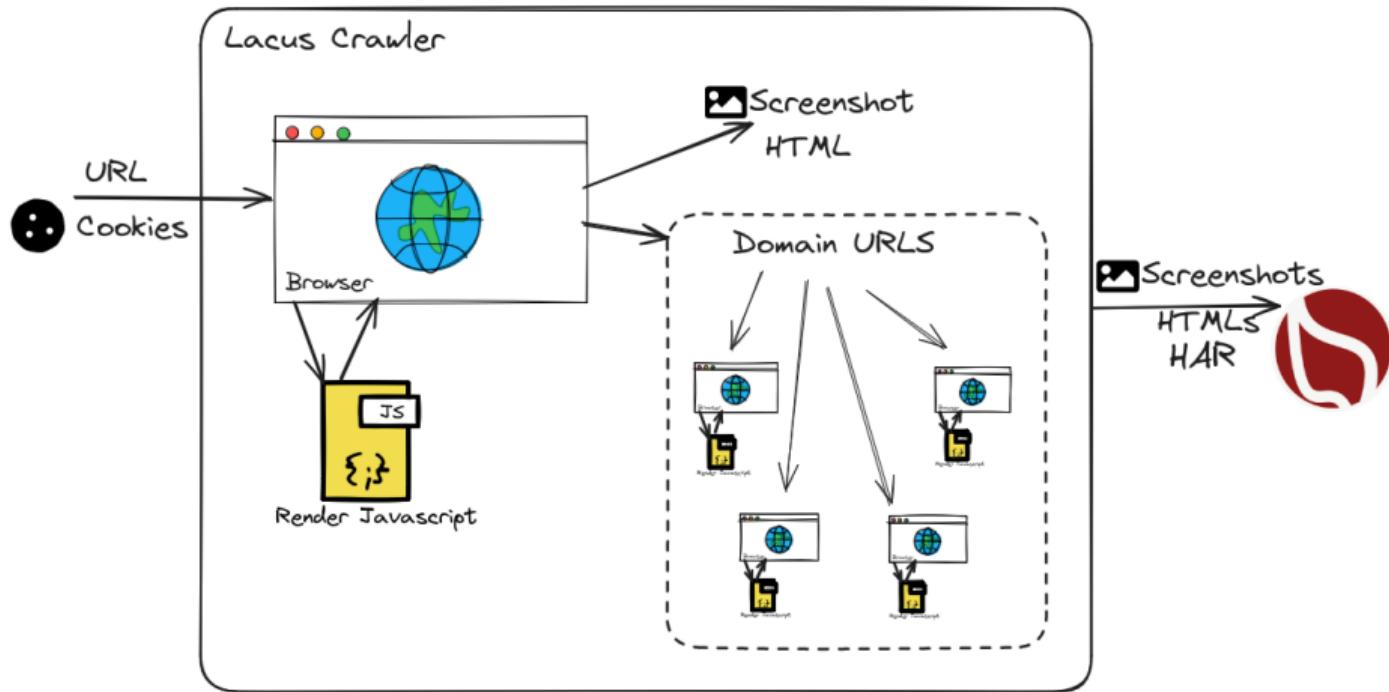
000201010211021643327300096494304155290090009649415312676076426760764000

Mai62360520AP9J3l0000000000000000708447583166304F69D

Bar Codes



Collection - Lacus Crawler



Crawler

- Crawlers are used to navigate on regular website as well as .onion addresses (via automatic extraction of urls or manual submission)
- Lacus⁶ ("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)

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

Lacus

- Lacus⁷ 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,

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

Crawler Settings - Lacus

All Lacus Crawler

Connected

Lacus URL	http://lacus.circl.lu:7100
Edit	

Crawlers

It works!

- 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

Cookiejar

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

Edit Description Add Cookies

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

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

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

```
{  
  "name": "remember_token",  
  "value": "12|58ddd151id74d341f23"  
}
```

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

Crawler: Cookiejar

3thxemke2x7hcibu.onion :

X 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 - 12:10:40 PORT: 80

Show 10 entries Search:

[Crawled Pastes](#)

Hide Full resolution

Shere Khan

Welcome back, zulipori. 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) Search

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
Inbox
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
Build/Ignore List
Manage Attachments
Save Drafts
Subscribed Threads
Forum Subscriptions
View Profile

Inbox Enter Keywords Search PMs Advanced Search

Message Title	Sender	Date/Time Sent (est)
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

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

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/mqbxyzjl4ladgz5cd.onion0aa31681-fa45-4fc3-8151-7a7c5ac7e906

Show Domain Correlations 2

CRYPTOCURRENCIES 2

Hide Full resolution

HOME ABOUT PROOF PRICE PAYMENT

DDOSTECH
WICKR: DDOS.TECHNOLOGY



Reviews

April 25, 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, 2018

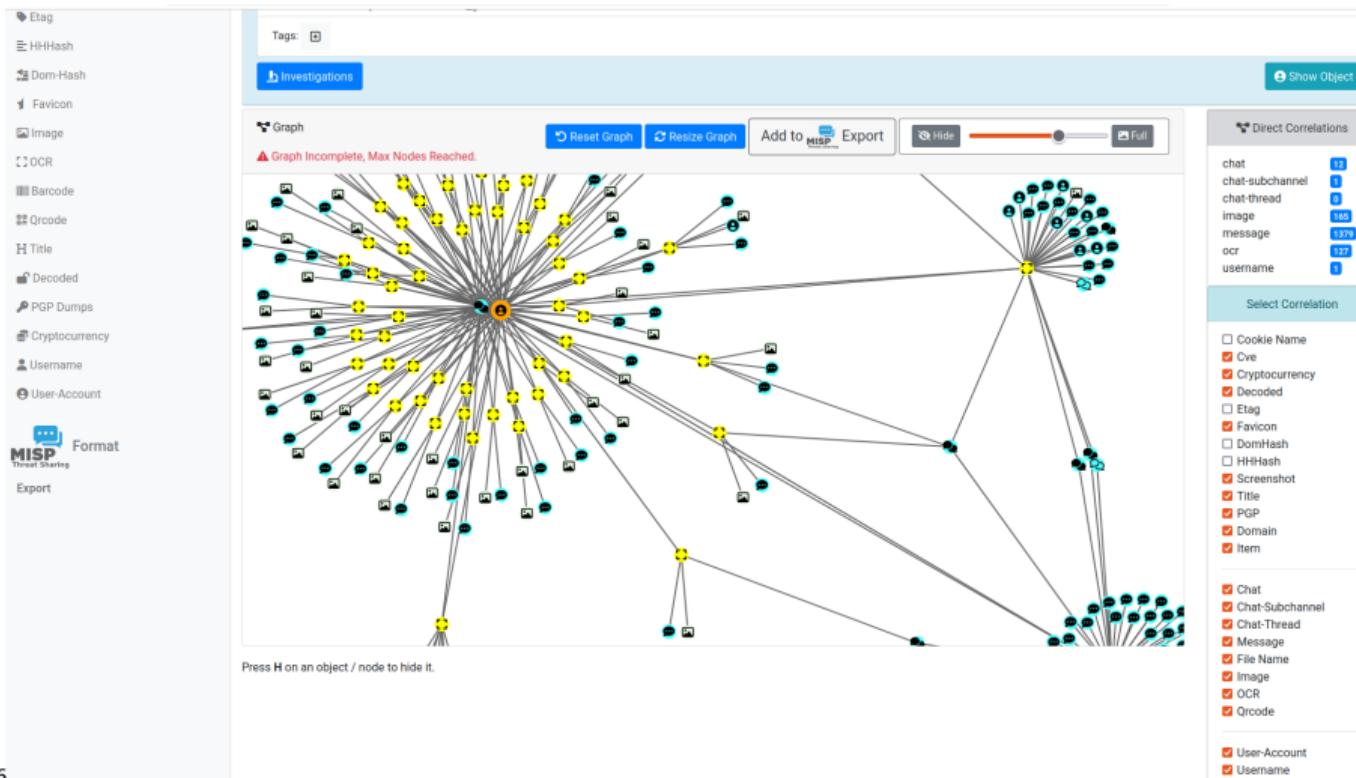
I found this site through YAHOO, immediately contacted this service, and I had a free attack for almost ten minutes.

We accept:

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



Correlations and relationship



Investigations

Tor Coin Mixer

UUID	9189d0e7c04c47a29f85666e9507e0a5
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

[Delete](#) [Edit](#) [Export as Event](#)

Objects			
Show 10 + entries			Search:
onion	jambler72pxlnhjmj3mhfdajmyddpxbuhfvoa32h5w4ctux3crqd.onion		infleak-automatic-detection:weak infleak-automatic-detection:pgp-public-key-block
onion	bitmaxht4cpndtuhwfuzzk23tvowswe4trdree74oxjmz2yyqqd.onion		infleak-automatic-detection:weak
key	0xD8B280956F0E7CAF		
mail	support@jambler.io		
telegram	jambler		

Dashboard

all project

Home Submit Tags Leaks Hunter Crawlers Objects Server Management Log Out

Feeders

2024 - 12 - 07

Category	Count
Barcodes	0
Chats	85
Cookie-Names	438
Cryptocurrencies	641
Cves	13
Decodeds	1902
DomHashes	8499
Domains	1580
Etags	5877
Favicons	778
File-Names	64
HHHashes	989
Images	434
Ocrs	198
PGP Dumps	144
Qrcodes	111
Titles	6895
User-Accounts	337
Usernames	498

Up: 1576, Down: 3197, Crawled: 4582, Queue: 14896

UP: 4, DOWN: 0, Crawled: 4, Queue: 0

tracker	Time
yara	Track drug market
yara	Track drug market

ID	Tags	Time
crawled/2024/12/07/ywyh44v7xxocqvuelfc68339-40a5-937e-e34b7ae351fb	infoleak-submission*crawler jxwjl ac2	16:26:01

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		Correlations Graph
 Extracted [?] [x]				
Show 10 ▼ entries		Search: <input type="text"/>		
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		

Example: Items Domain

Crawler

Last Origin:

- crawled/2025/02/03/drugsxeuauw4xt6eoqwztfkpa2ul5453q4ao4kqrtdhfoy6jmu5ryd.onionae52293f-ffda-4bd9-b037
- drugsxeuauw4xt6eoqwztfkpa2ul5453q4ao4kqrtdhfoy6jmu5ryd.onion

url <http://drugsxeuauw4xt6eoqwztfkpa2ul5453q4ao4kqrtdhfoy6jmu5ryd.onion/?product=chip-skimmer>

The screenshot shows a web browser window displaying an onion website for an 'Anonymous Marketplace'. The URL is <http://drugsxeuauw4xt6eoqwztfkpa2ul5453q4ao4kqrtdhfoy6jmu5ryd.onion/?product=chip-skimmer>. The page has a blue header with the text 'Anonymous Marketplace' and a search bar. Below the header, there's a navigation menu with links like 'Homepage', 'Carding', 'Hacking Services', 'Counterfeits / BankNotes', 'Money transfers', and 'Reviews'. A shopping cart icon indicates a total value of '\$13,702.00'. The main content area features a product listing for a 'CHIP MSR CARD SKIMMER MCR 200 EMV MAG STRIPE READER WRITER'. The product image shows various electronic components, including a grey box, a black power adapter, and several cables. The price is listed as '\$249.00' with a strike-through line and a new price of '\$149.00'. An 'Add to cart' button is visible. The product is categorized under 'hardwares, other services'. At the bottom of the page, 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 Reader Writer Magnetic Stripe Card Reader Writer & EMV chip Encoder Track 1,2,3
Covers with EMV Reader/Writer Software v8.6 for Windows 95/98/Me/2000/XP/Vista/Windows 7/Windows 8/Windows 8.1/Windows 10/MacOS/Linux (Only Ubuntu and Debian)
1. Magnetic Card Reader/Writer HCC206

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
SchifFrvi:102091
Chaos84:HOLES244
Riptor795:blade7
Dom180:harkonen
Casper1112:superhero
```

Example: Search by tags

Search Items by Tags :

Date Range: 2023-05-14 to 2023-05-27

Tags: infoleak:automatic-detection="cve" x infoleak:automatic-detection="bitcoin-address" x

Search Items

Show 10 entries

Search:

Date	Item	Action
2023/05/16	archive/gist.github.com/2023/05/16/Vazgen7788_c036ee7aad316d9038f2a3968abbcc5d.gz infoleak:automatic-detections="searchsploit-tool" infoleak:automatic-detection="cve" infoleak:automatic-detection="ethereum-address" infoleak:automatic-detection="base64" infoleak:automatic-detection="bitcoin-address"	
2023/05/16	archive/gist.github.com/2023/05/16/vijay922_d35cf2f5c9abe682140379e35d5cd935.gz infoleak:automatic-detections="searchsploit-tool" infoleak:automatic-detection="cve" infoleak:automatic-detection="ethereum-address" infoleak:automatic-detection="base64" infoleak:automatic-detection="bitcoin-address"	
2023/05/16	archive/gist.github.com/2023/05/16/DmitriyLewen_930515cde810283b7804950efafe3273.gz infoleak:automatic-detection="searchsploit-tool" infoleak:automatic-detection="cve" infoleak:automatic-detection="credential" infoleak:automatic-detection="bitcoin-address"	
2023/05/19	archive/gist.github.com/2023/05/19/GrahamcOfBorg_46422a069e8b942352a65f3121a769c5.gz infoleak:automatic-detection="cve" infoleak:automatic-detection="credential" infoleak:automatic-detection="bitcoin-address"	
2023/05/26	archive/pastebin.com_pro/2023/05/26/SewhAHi0.gz infoleak:automatic-detections="ethereum-address" infoleak:automatic-detection="cve" infoleak:automatic-detection="bitcoin-address"	

Showing 1 to 5 of 5 entries

Previous 1 Next

MISP

MISP Taxonomies

- **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 ⁸.
- Can be easily cherry-picked or extended

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

Taxonomies useful in AIL

- **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
- **fpf⁹**: Evaluate the degree of identifiability of personal data and the types of pseudonymous data, de-identified data and anonymous data.

⁹Future of Privacy Forum

Taxonomies useful in AIL

- **tor**: Describe Tor network infrastructure.
- **dark-web**: Criminal motivation on the dark web.
- **copine-scale¹⁰**: Categorise the severity of images of child sex abuse.

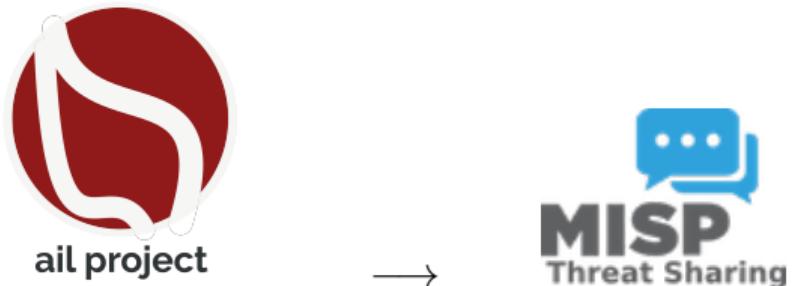
¹⁰Combating Paedophile Information Networks in Europe

threat sharing and incident response platforms



Goal: submission to threat sharing and incident response platforms.

threat sharing and incident response platforms



1. Use infoleak taxonomy¹¹
2. Add your own tags
3. Export AIL objects to MISP core format
4. Download it or Create a MISP Event¹²

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

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

MISP Export

1Gt545E48EPsyTC8voKQDCFfpTkwiuXduw :

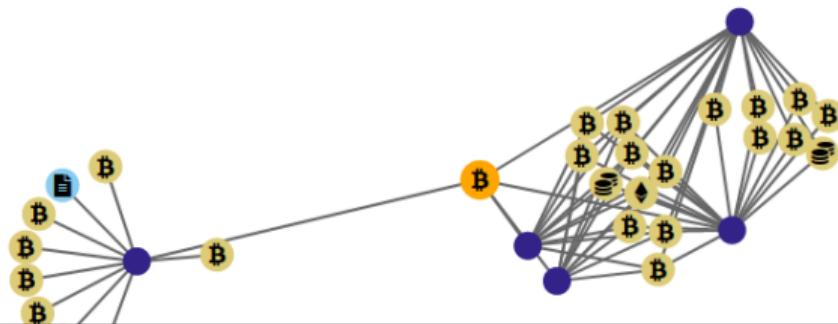
Object type	type	First seen	Last seen	Nb seen
cryptocurrency	 bitcoin	2020/01/17	2020/02/20	5

Expand Bitcoin address

Graph

Resize Graph

Add to MISP Threat Sharing Export



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](#)

[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

[Forgot your password?](#)

What's the best way to... ?

MISP Export

 MISP Exporter
Threat Sharing

Select a list of objects to export

Object Type	Object ID	Lvl	
Object type...		0	<input data-bbox="1301 404 1355 435" type="button" value="+"/>
Object type...	1Gt545E48EPsyTC8voKQDCFfpTkwiuXduw	1	<input checked="" data-bbox="1301 445 1355 476" type="button" value="Delete"/>
Domain	nttfj36sp47cw2yecop572zjvjeazgazieunllouudplzqt2m5h465yd.onion	0	<input checked="" data-bbox="1301 486 1355 517" type="button" value="Delete"/>

JSON Export Export to MISP Instance

Distribution:

Threat Level:

Analysis:

Event Info:

Publish Event

Automatic MISP Export on tags

MISP Auto Event Creation Enabled



× Disable Event Creation

MISP Tags To Push : 3 / 89

Show	10	entries
Enabled	11	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 Auto Alert Creation Disabled



Enable Alert Creation

The Hive Tags To Push : 4 / 89

Show	10	entries
Enabled	11	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"

API

AIL exposes a ReST API which can be used to interact with the back-end¹³.

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

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

Setting up the framework

Setting up AIL-Framework from source

Setting up AIL-Framework from source

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

Feeding the framework

Feeding Data to AIL

There are different ways to feed data into AIL:

1. AIL Importers:
 - o Dir / Files
 - o ZMQ
 - o *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)

Feeding Data to AIL - Limitation



/!\\ 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

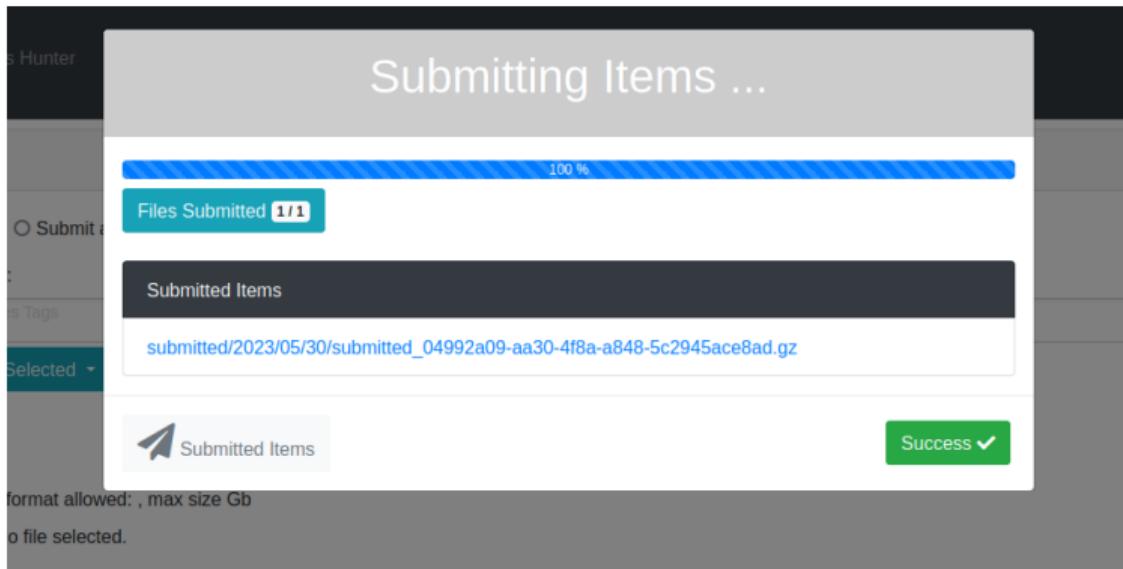
Via the UI (1)

The screenshot shows the 'Submit Item' page of a web application. 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 'Toggle Sidebar' button and a 'Submit Items' section. The main content area is titled 'Submit Item' and contains the following fields:

- Submit a file Submit a text
- Optional Tags:
 - Add Taxonomies Tags
 - Taxonomic Selected
- Add Galaxies Tags
- Galaxy Selected
- Submit a text, max size 1.0 Mb
- Source
- test text to submit

At the bottom is a large blue 'Submit Item' button.

Via the UI (2)



API - Feeding AIL with your own data

api/v1/import/item

```
1 {
2     "type": "text",
3     "tags": [
4         "infoleak:analyst-detection=\"private-key\""
5     ],
6     "text": "text to import"
7 }
```

Importers

- Importers are located in the `/bin/importer` directory
- They are used to import different types of data into AIL
- Adding new Importers is straightforward.
- Available Importers:
 - AIL Feeders
 - ZMQ
 - pystemon
 - Files

File Importer

- importer/FileImporter.py

Import File

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

Import Dir

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

AIL feeders Importers

- **12+ feeders are available** for all AIL users to feed from external sources
- External feeders can run anywhere and are completely separated from AIL framework
- The feeder can use their **own internal logic** and even push JSON metadata
- Feeder are then pushing the generated JSON to AIL API

Certificate transparency feeder for AIL

- ail-feeder-cti¹⁴ is a generic software to extract information from a certstream server (certificate transparency)
- All metadata extracted will be processed by AIL
- Onion addresses crawled automatically by AIL if seen in a certificate

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

GitHub archive and GitHub repository

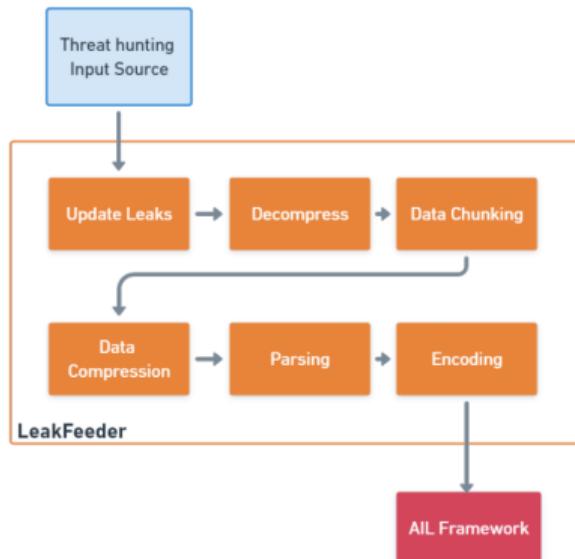
- ail-feeder-gharchive¹⁵ is a generic software to extract informations from **GHArchive**, collect and feed AIL via AIL ReST API
- ail-feeder-github-repo¹⁶ is collecting from a GitHub repository and push everything to AIL
- For monitoring a set of **suspicious git repositories** or finding leaks on existing or managed git repositories, it's a simple way to feed AIL with such source.

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

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

AIL LeakFeeder

- ail-feeder-leak¹⁷ automates the process to feed leaked large files automatically to AIL



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

AIL feeder ActivityPub

- ail-feeder-activity-pub¹⁸ is feeder for the ActivityPub standard used in distributed social networks
- Accounts are required on the ActivityPub instance to get the stream

¹⁸<https://github.com/ail-project/ail-feeder-activity-pub>

AIL feeder telegram

- ail-feeder-telegram¹⁹ is a **Telegram feeder**
- An API ID/hash for Telegram is required and linked to your Telegram phone number

¹⁹<https://github.com/ail-project/ail-feeder-telegram>

More feeders

- ail-feeder-discord²⁰ is a generic **Discord** feeder for AIL
- ail-feeder-atom-rss²¹ is an **Atom and RSS reader** and feeder for AIL
- ail-feeder-jsonlogs²² is a **JSON aggregator** to submit generic JSON input into AIL

²⁰<https://github.com/ail-project/ail-feeder-discord>

²¹<https://github.com/ail-project/ail-feeder-atom-rss>

²²<https://github.com/ail-project/ail-feeder-jsonlogs>

Feeding AIL with custom JSON



conti leaks
@ContiLeaks

...

conti jabber leaks 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

```
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'] = "{}".format(uuid.uuid4())
    pyail.feed_json_item(tmp, tmpmt, ailfeedertype, source_uuid)
```

feeder.py

```
$ 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:
 - PGP keys

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

- onion hidden services | grep http:// |
- telegram addresses | grep tg:// |
- bitcoins addresses | 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 AIL

```
1  
2 # Launch the system and the web interface  
3 cd bin/  
4 ./LAUNCH -l
```

Updating AIL

Launch the updater:

```
1 cd bin/
2 # git pull and launch all updates:
3 ./LAUNCH -u
4
5
6 # PS:
7 # The Updater is launched by default each time
8 # you start the framework with
9 # ./LAUNCH -l
```

Running your own instance using the virtual machine

Login and passwords:

```
1 # Web interface (default network settings)
2   https://127.0.0.1:7000/
3 # Web interface:
4   admin@admin.test
5   Password1234
6 # SSH:
7   test
8   Password1234
```

AIL ecosystem - Challenges and design

AIL ecosystem: Technologies used

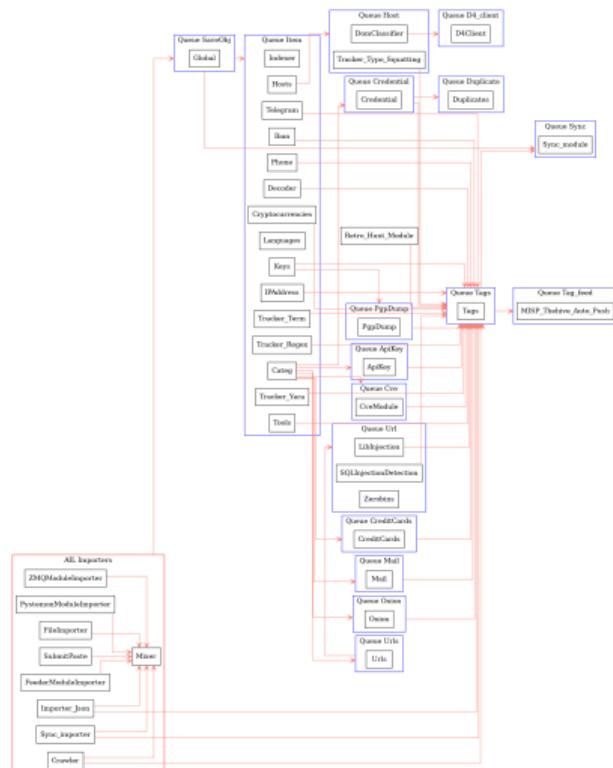
Programming language: Full python3

Databases: Redis and Kvrocks

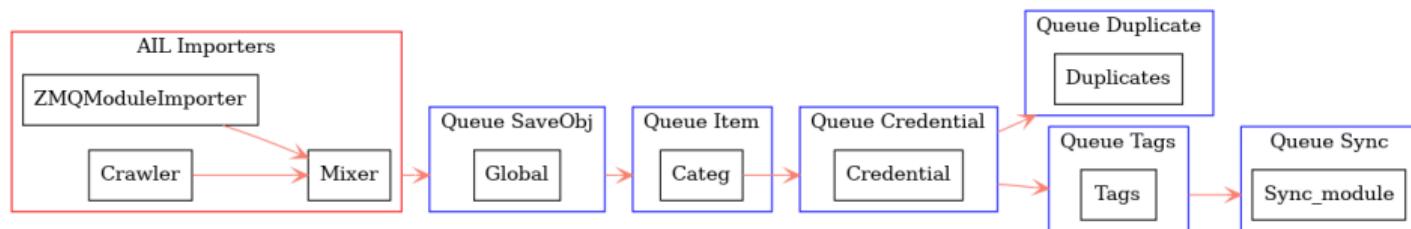
Server: Flask

Data message passing: Redis Set

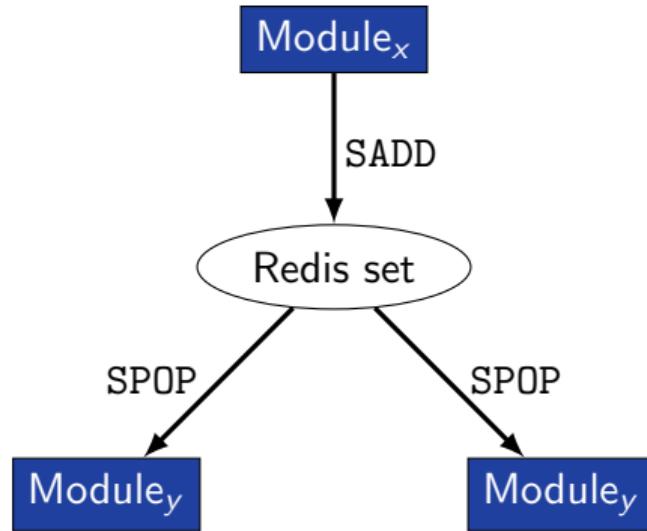
AIL global architecture: Data streaming between module



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



Message consuming

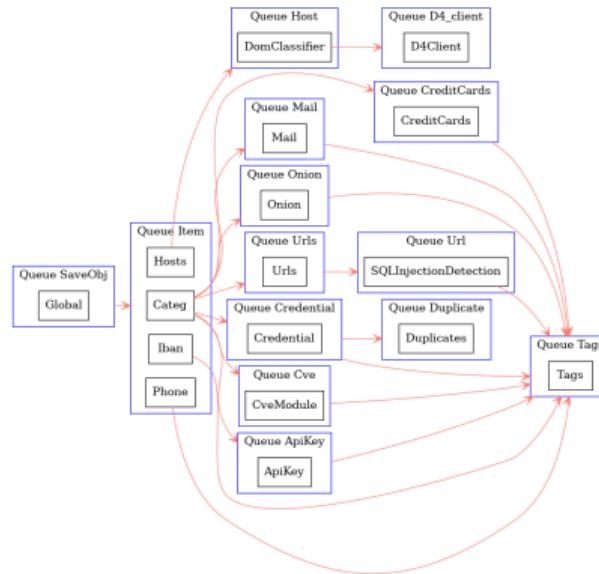


- No message lost nor double processing
- Multiprocessing!

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

Writing your own modules - /bin/modules/TemplateModule.py

```
1 from modules.abstract_module import AbstractModule
2
3 class NewModule(AbstractModule):
4
5     def __init__(self):
6         super().__init__()
7         self.logger.info(f'Module {self.module_name} initialized')
8
9     # Do something with the message from the queue
10    def compute(self, message):
11        # Process Message
12
13    # LAUNCH MODULE
14    if __name__ == '__main__':
15        module = NewModule()
16        module.run()
17
18
```

Writing your own Importer - /bin/importer/

```
1 from importer.abstract_importer import AbstractImporter
2 from modules.abstract_module import AbstractModule
3
4 class MyNewImporter(AbstractImporter):
5
6     def __init__(self):
7         super().__init__()
8         # super().__init__(queue=True)    # if it's an one-time run importer
9         self.logger.info(f'Importer {self.name} initialized')
10
11    def importer(self, my_var): # import function
12        # Process my_var and get content to import
13        content = GET_MY_CONTENT_TO_IMPORT
14        # if content is not gzipped and/or not b64 encoded,
15        # set gzipped and/or b64 to False
16        message = self.create_message(item_id, content)
17        return message
18        # if it's an one-time run, otherwise create an AIL Module
19        # self.add_message_to_queue(message)
20
21 class MyNewModuleImporter(AbstractModule):
22     def __init__(self):
23         super().__init__() # init module ...
24         # init module ...
```

Writing your own Importer - /bin/importer/

```
1
2     def get_message(self):
3         return self.importer.importer()
4
5     def compute(self, message):
6         self.add_message_to_queue(message)
7
8 if __name__ == '__main__':
9     module = MyNewModuleImporter()
10    module.run()
11
12    # if it's an one-time run:
13    # importer = MyImporter()
14    # importer.importer(my_var)
15
16
```

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!



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.

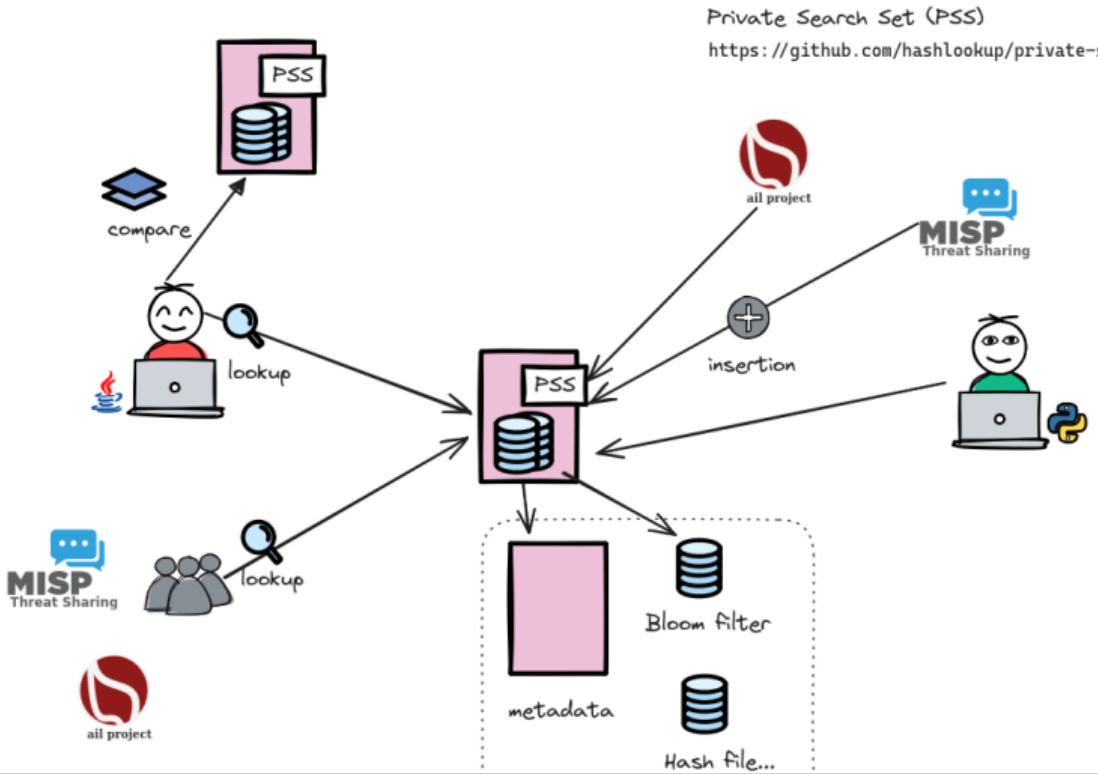
Implementation Steps in AIL project

- **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

Private Search Set (PSS)



Annexes

Managing AIL: 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