AIL Framework for Analysis of Information Leaks

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

CIRCL
Computer Incident
Response Center
Luxembourg

Alexandre Dulaunoy
alexandre.dulaunoy@circl.lu
Aurelien Thirion
aurelien.thirion@circl.lu
Jean-Louis Huynen
jean-louis.huynen@circl.lu

info@circl.lu

April 1, 2021

Links

- AIL project https://github.com/ail-project
- AIL framework https://github.com/ail-project/ail-framework
- Training materials https://github.com/ail-project/ail-training
- Online chat https://gitter.im/ail-project/community

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 regulation[1].

[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.

Objectives

Our objectives

- Show how to use and extend an open source tool to monitor web pages, pastes, forums and hidden services
- Explain challenges and the design of the AIL open source framework
- Learn how to create new modules
- Learn how to use, install and start AIL
- **Supporting investigation using the AIL framework**

AIL Framework

From a requirement to a solution: AIL Framework

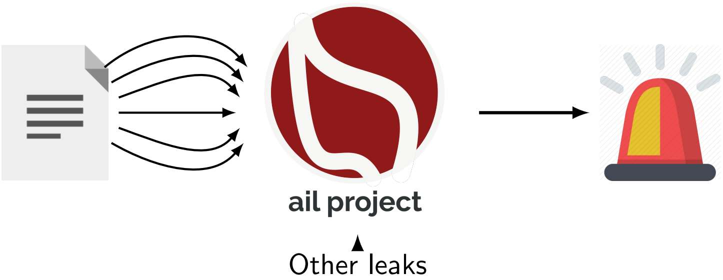
History:

- 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**[2].

[2] https://github.com/ail-project/

AIL Framework: A framework for Analysis of Information Leaks

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



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 websites

Support CERT and Law Enforcement activities

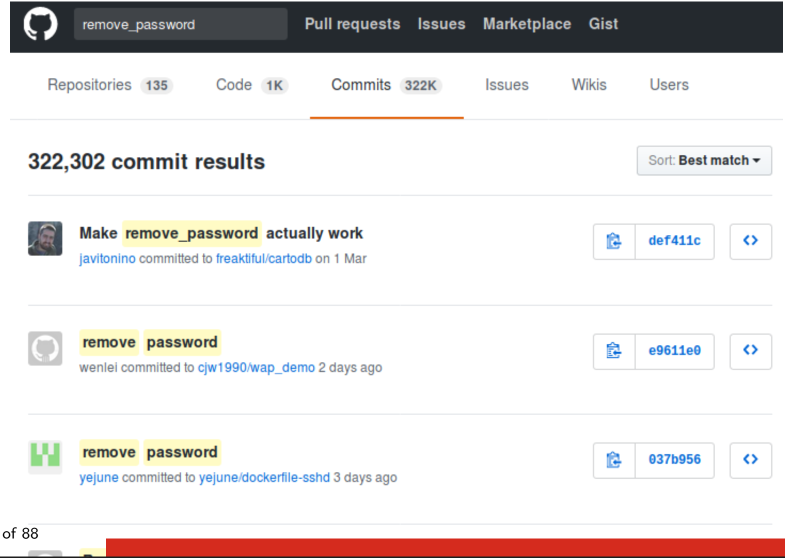
- Proactive investigation: leaks detection
  - List of emails and passwords
  - Leaked database
  - AWS Keys
  - Credit-cards
  - PGP private keys
  - Certificate private keys
- Feed Passive DNS or any passive collection system
- CVE and PoC of vulnerabilities most used by attackers

Support CERT and Law Enforcement activities

- Website monitoring
  - monitor booters
  - Detect encoded exploits (WebShell, malware encoded in Base64, ...)
  - SQL injections
- Automatic and manual submission to threat sharing and incident response platforms
  - MISP
  - TheHive
- Term/Regex/YARA monitoring for local companies/government

Sources of leaks

Mistakes from users:



remove_password

322,302 commit results

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

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

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

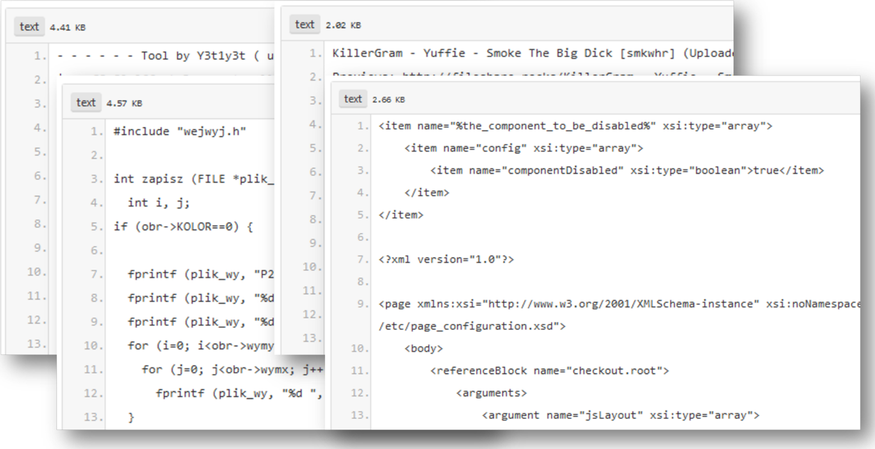
Sources of leaks: Paste monitoring

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

Sources of leaks: Paste monitoring

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

Examples of pastes (items)



text 4.41 KB

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

text 4.57 KB

```
#include "wejwyj.h"

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

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

text 2.02 KB

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

text 2.66 KB

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

<?xml version="1.0"?>

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

Why so many leaks?

- Economical interests (e.g. Adversaries promoting services)
- Ransom model (e.g. To publicly pressure the victims)
- Political motives (e.g. Adversaries showing off)
- Collaboration (e.g. Criminals need to collaborate)
- Operational infrastructure (e.g. malware exfiltrating information on a pastie website)
- Mistakes and errors

Are leaks frequent?

Yes!

and we have to deal with this as a CSIRT.

- **Contacting companies or organisations** who did specific accidental leaks
- **Discussing with media** about specific case of leaks and how to make it more practical/factual for everyone
- Evaluating the economical market for cyber criminals (e.g. DDoS booters[3] or reselling personal information - reality versus media coverage)
- Analysing collateral effects of malware, software vulnerabilities or exfiltration

→ And it's important to detect them automatically.

[3] `https://github.com/D4-project/`

Paste monitoring at CIRCL: Statistics

- Monitored paste sites: 27
  - *gist.github.com*
  - *ideone.com*
  - *...*

	2016	2017	08.2018
Collected pastes	18,565,124	19,145,300	11,591,987
Incidents	244	266	208

Table: Pastes collected and incident[4] raised by CIRCL

[4] `http://www.circl.lu/pub/tr-46`

Current capabilities

AIL Framework: Current capabilities

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

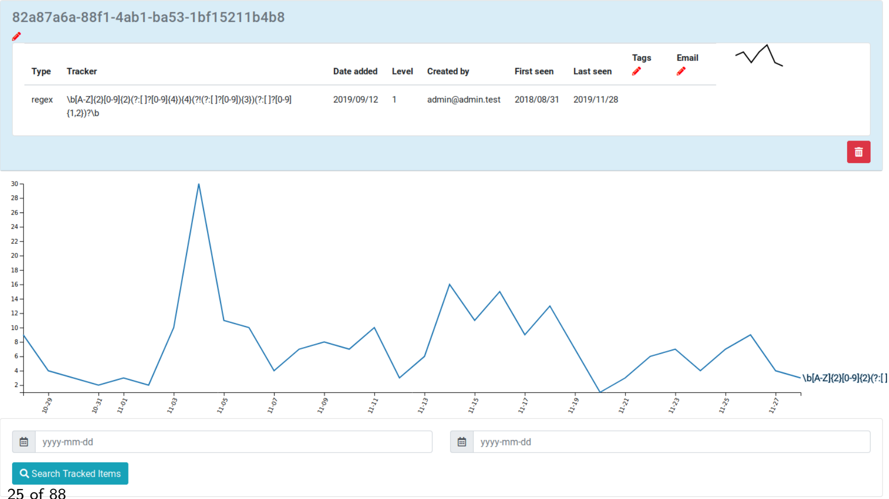
AIL Framework: Current features

- Extracting **credit cards numbers, credentials, phone numbers, ...**
- Extracting and validating potential **hostnames**
- Keeps track of **duplicates**
- Submission to threat sharing and incident response platform (**MISP** and **TheHive**)
- **Full-text indexer** to index unstructured information
- **Tagging** for classification and searches
- Terms, sets, regex and YARA **tracking and occurences**
- Archives, files and raw **submission** from the UI
- PGP, Cryptocurrency, Decoded (Base64, ...) and username Correlation
- And many more

Terms Tracker

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

Terms Tracker:



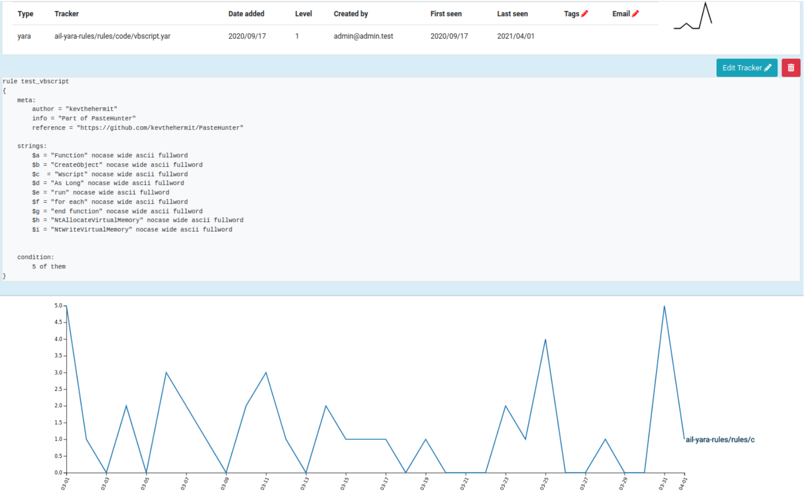
82a87a6a-88f1-4ab1-ba53-1bf15211b4b8

Type	Tracker	Date added	Level	Created by	First seen	Last seen	Tags	Email
regex	\b[A-Z]{2}[0-9]{2}(?:[ ]?[0-9]{4}){4}(?!(?:[ ]?[0-9]){3})(?:[ ]?[0-9]{1,2})?\b	2019/09/12	1	admin@admin.test	2018/08/31	2019/11/28		

yyyy-mm-dd

yyyy-mm-dd

YARA Tracker:

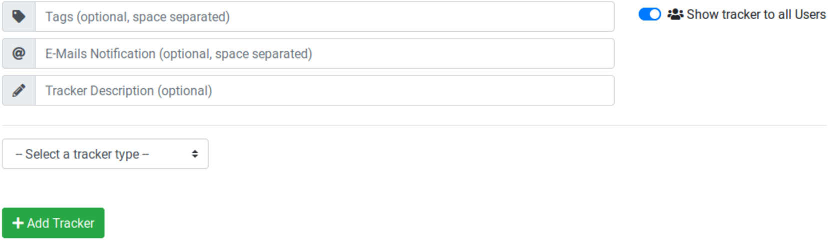


Type	Tracker	Date added	Level	Created by	First seen	Last seen	Tags	Email
yara	ail-yara-rules/rules/code/vbscript.yar	2020/09/17	1	admin@admin.test	2020/09/17	2021/04/01		

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

    strings:
        $a = "Function" nocase wide ascii fullword
        $b = "CreateObject" nocase wide ascii fullword
        $c  = "Wscript" nocase wide ascii fullword
        $d = "As Long" nocase wide ascii fullword
        $e = "run" nocase wide ascii fullword
        $f = "for each" nocase wide ascii fullword
        $g = "end function" nocase wide ascii fullword
        $h = "NtAllocateVirtualMemory" nocase wide ascii fullword
        $i = "NtWriteVirtualMemory" nocase wide ascii fullword


    condition:
        5 of them
}
```

Terms Tracker - Practical part

- **Create and test** your own tracker



Tags (optional, space separated)

E-Mails Notification (optional, space separated)

Tracker Description (optional)

– Select a tracker type –

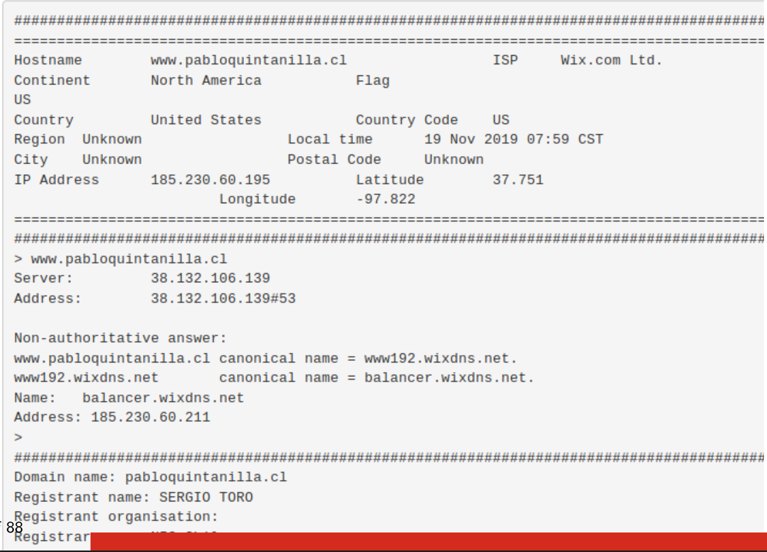
Show tracker to all Users

+ Add Tracker

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

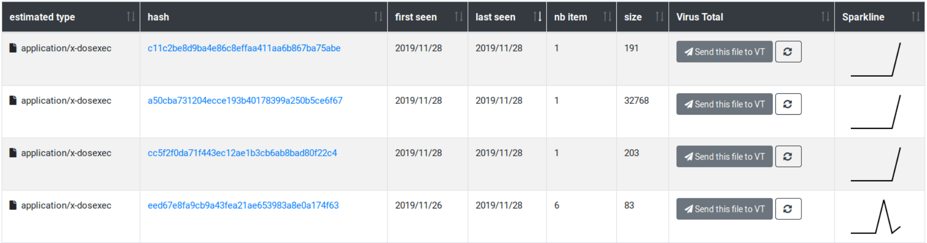


```
########################################################################################
========================================================================================
Hostname        www.pabloquintanilla.cl                 ISP     Wix.com Ltd.
Continent       North America           Flag
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
```

Decoder

- Search for encoded strings
  - Base64
  - Hexadecimal
  - Binary
- Guess Mime-type
- Correlate paste with decoded items

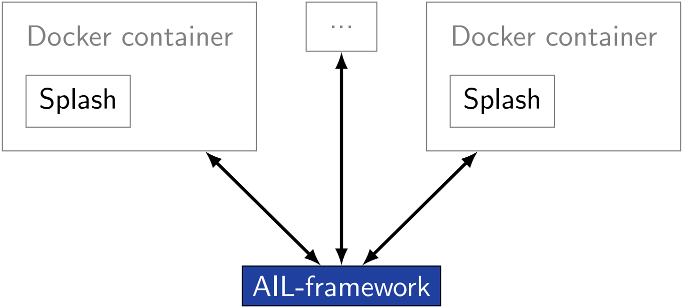
Decoder:



estimated type	hash	first seen	last seen	nb item	size	Virus Total	Sparkline
application/x-dosexec	c11c2be8d9ba4e86c8effaa411aa6b867ba75abe	2019/11/28	2019/11/28	1	191	Send this file to VT	
application/x-dosexec	a50cba731204ecce193b40178399a250b5ce6f67	2019/11/28	2019/11/28	1	32768	Send this file to VT	
application/x-dosexec	cc5f2f0da71f443ec12ae1b3cb6ab8bad80f22c4	2019/11/28	2019/11/28	1	203	Send this file to VT	
application/x-dosexec	eed67e8fa9cb9a43fea21ae653983a8e0a174f63	2019/11/26	2019/11/28	6	83	Send this file to VT	

Crawler

- Crawlers are used to navigate on regular website as well as .onion addresses (via automatic extraction of urls or manual submission)
- Splash (”scriptable” browser) is rending the pages (including javascript) and produce screenshots (HAR archive too)



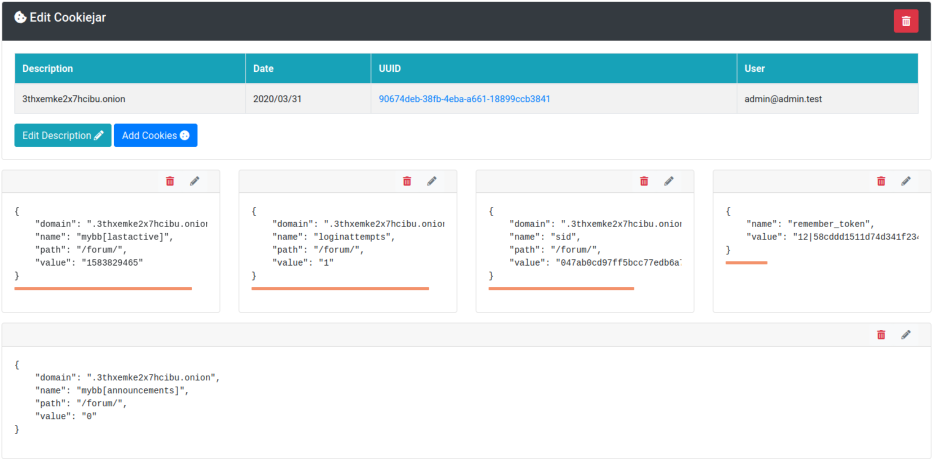
Crawler

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 $= 1$)

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

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

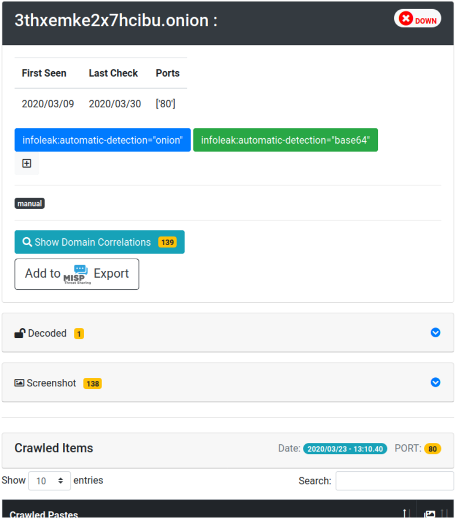
```
{
    "domain": ".3thxemke2x7hcibu.onio
    "name": "loginattempts",
    "path": "/forum/",
    "value": "1"
}
```

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

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

```
{
    "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']

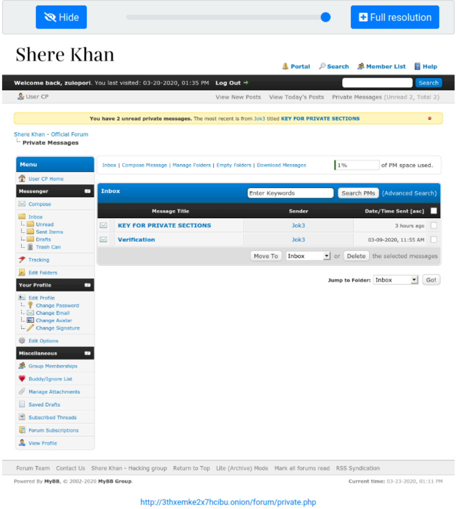
infoleak:automatic-detection="onion" infoleak:automatic-detection="base64"

manual

Crawled Items Date: 2020/03/23 - 13:10.40 PORT: 80

Show 10 entries Search:

Crawled Pastes



Shere Khan

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

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

1% of PM space used.

Inbox

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

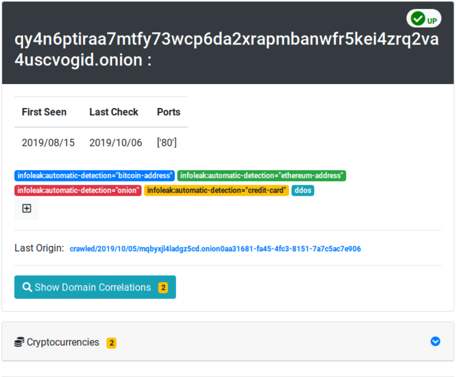
Jump to Folder: Inbox Go!

Powered By **MyBB**, © 2002-2020 **MyBB Group**.

Current time: 03-23-2020, 01:11 PM

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

Crawler: DDoS Booter



UP

qy4n6ptiraa7mtfy73wcp6da2xrapmbanwfr5kei4zrq2va4uscvogid.onion :

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

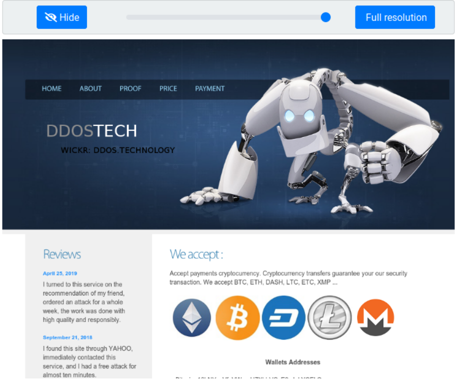
infoleak:automatic-detection="bitcoin-address" infoleak:automatic-detection="ethereum-address" infoleak:automatic-detection="onion" infoleak:automatic-detection="credit-card" ddos

⊞

Last Origin: crawled/2019/10/05/mqbyxjl4ladgz5cd.onion0aa31681-fa45-4fc3-8151-7a7c5ac7e906

Show Domain Correlations 2

Cryptocurrencies 2



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 responsibly.

September 21, 2018

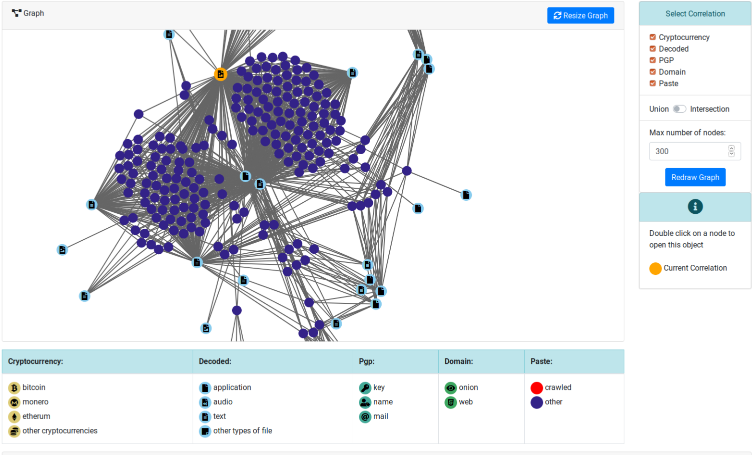
I found this site through YAHOO, immediately contacted this service, and I had a free attack for almost ten minutes.

We accept :

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

Wallets Addresses

Correlations and relationship



Select Correlation

- ☑ Cryptocurrency
- ☑ Decoded
- ☑ PGP
- ☑ Domain
- ☑ Paste

Union ○ Intersection

Max number of nodes: 300

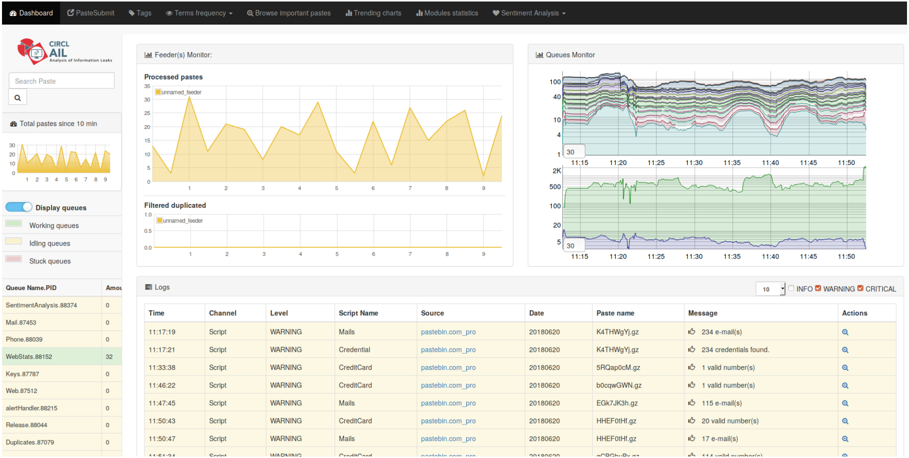
Double click on a node to open this object

● Current Correlation

Cryptocurrency:	Decoded:	Pgp:	Domain:	Paste:
bitcoin	application	key	onion	crawled
monero	audio	name	web	other
etherum	text	mail		
other cryptocurrencies	other types of file			

Live demo!

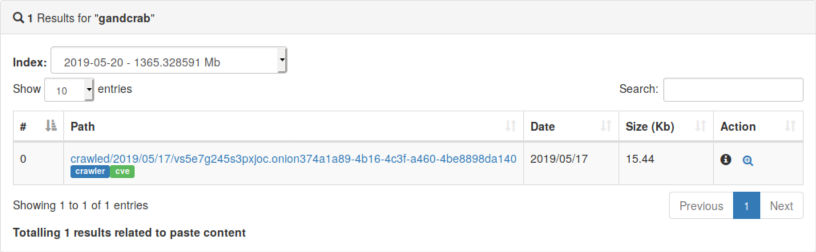
Example: Dashboard



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

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

Example: Text search



1 Results for "**gandcrab**"

Index: 2019-05-20 - 1365.328591 Mb

Show 10 entries

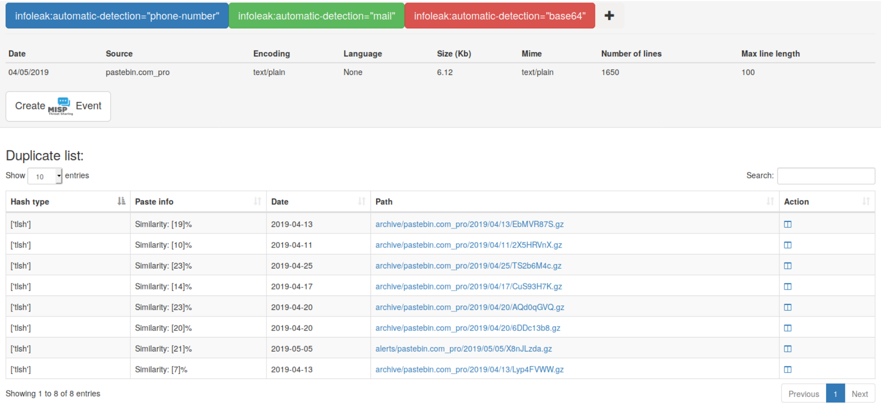
Search:

#	Path	Date	Size (Kb)	Action
0	crawled/2019/05/17/vs5e7g245s3pxjoc.onion374a1a89-4b16-4c3f-a460-4be8898da140 crawler cve	2019/05/17	15.44	ℹ 🔍

Showing 1 to 1 of 1 entries

Totalling 1 results related to paste content

Example: Items Metadata (1)



infoleak:automatic-detection="phone-number" infoleak:automatic-detection="mail" infoleak:automatic-detection="base64" +

Date	Source	Encoding	Language	Size (Kb)	Mime	Number of lines	Max line length
04/05/2019	pastebin.com_pro	text/plain	None	6.12	text/plain	1650	100

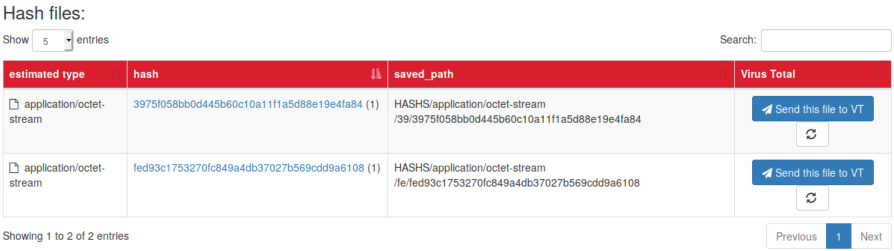
Duplicate list:

Show 10 entries Search:

Hash type	Paste info	Date	Path	Action
['tlsh']	Similarity: [19]%	2019-04-13	archive/pastebin.com_pro/2019/04/13/EbMVR87S.gz	◫
['tlsh']	Similarity: [10]%	2019-04-11	archive/pastebin.com_pro/2019/04/11/2X5HRVnX.gz	◫
['tlsh']	Similarity: [23]%	2019-04-25	archive/pastebin.com_pro/2019/04/25/TS2b6M4c.gz	◫
['tlsh']	Similarity: [14]%	2019-04-17	archive/pastebin.com_pro/2019/04/17/CuS93H7K.gz	◫
['tlsh']	Similarity: [23]%	2019-04-20	archive/pastebin.com_pro/2019/04/20/AQd0qGVQ.gz	◫
['tlsh']	Similarity: [20]%	2019-04-20	archive/pastebin.com_pro/2019/04/20/6DDc13b8.gz	◫
['tlsh']	Similarity: [21]%	2019-05-05	alerts/pastebin.com_pro/2019/05/05/X8nJLzda.gz	◫
['tlsh']	Similarity: [7]%	2019-04-13	archive/pastebin.com_pro/2019/04/13/Lyp4FVWW.gz	◫

Showing 1 to 8 of 8 entries

Example: Items Metadata (2)



Hash files:

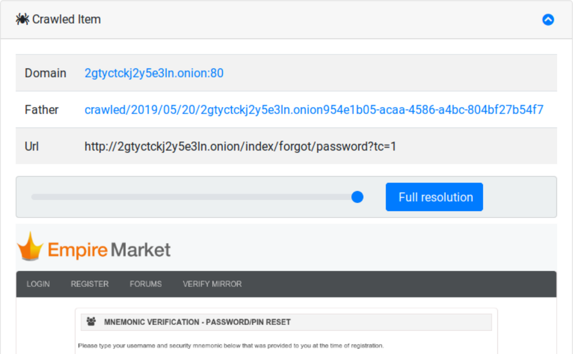
Show 5 entries

Search:

estimated type	hash	saved_path	Virus Total
application/octet-stream	3975f058bb0d445b60c10a11f1a5d88e19e4fa84 (1)	HASHS/application/octet-stream /39/3975f058bb0d445b60c10a11f1a5d88e19e4fa84	Send this file to VT
application/octet-stream	fed93c1753270fc849a4db37027b569cdd9a6108 (1)	HASHS/application/octet-stream /fe/fed93c1753270fc849a4db37027b569cdd9a6108	Send this file to VT

Showing 1 to 2 of 2 entries

Example: Items Metadata (3)



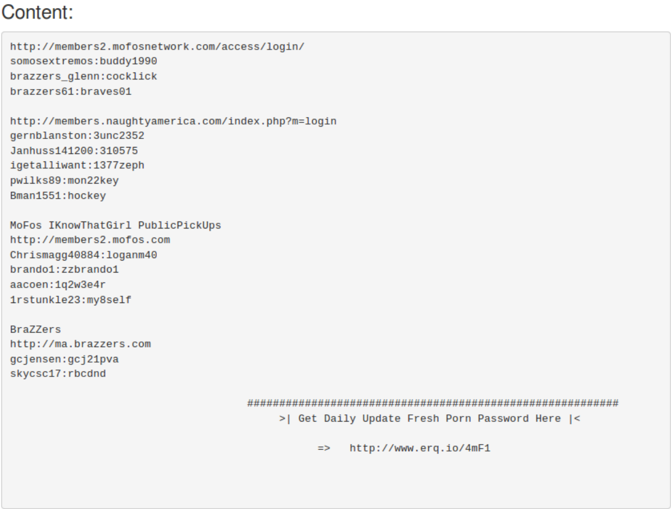
Crawled Item

Domain	2gtyctckj2y5e3ln.onion:80
Father	crawled/2019/05/20/2gtyctckj2y5e3ln.onion954e1b05-acaa-4586-a4bc-804bf27b54f7
Url	http://2gtyctckj2y5e3ln.onion/index/forgot/password?tc=1

MNEMONIC VERIFICATION - PASSWORD/PIN RESET

Please type your username and security mnemonic below that was provided to you at the time of registration.

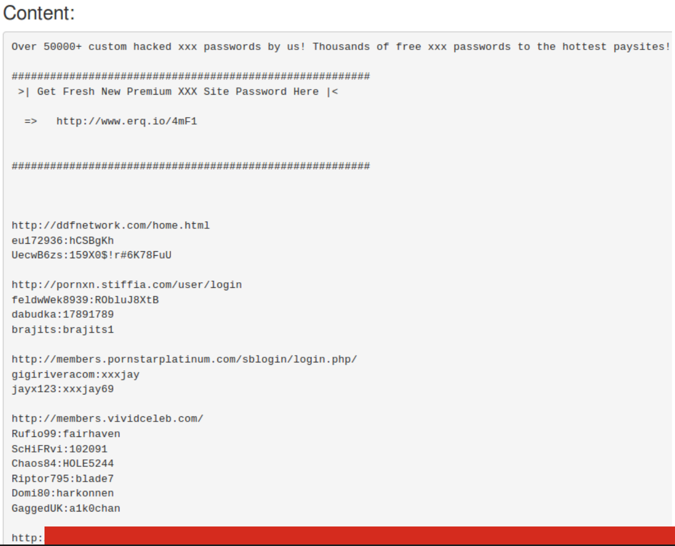
Example: Browsing content



Content:

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

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

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

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

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

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

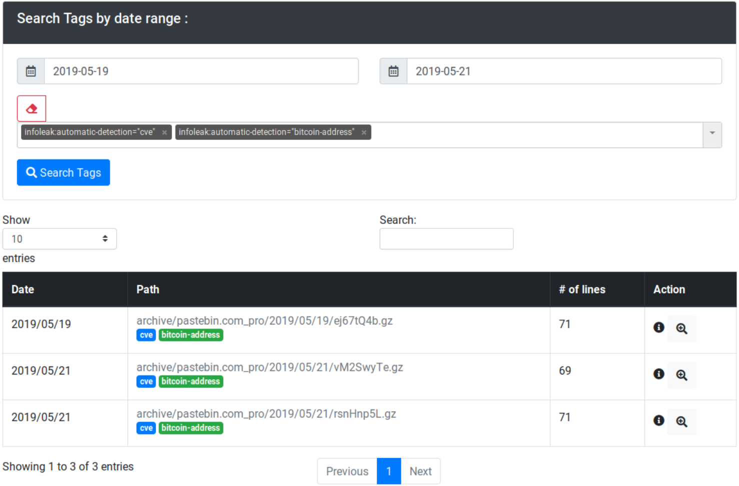
Example: Browsing content



Content:

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

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

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


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




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

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

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

http://members.vividceleb.com/
Rufio99:fairhaven
ScHiFRvi:102091
Chaos84:HOLE5244
Riptor795:blade7
Domi80:harkonnen
GaggedUK:a1k0chan

http:
```

Example: Search by tags



Search Tags by date range :

2019-05-19

2019-05-21

infoleak:automatic-detection="cve" × infoleak:automatic-detection="bitcoin-address" ×

Show
10
entries

Search:

Date	Path	# of lines	Action
2019/05/19	archive/pastebin.com_pro/2019/05/19/ej67tQ4b.gz cve bitcoin-address	71	
2019/05/21	archive/pastebin.com_pro/2019/05/21/vM2SwyTe.gz cve bitcoin-address	69	
2019/05/21	archive/pastebin.com_pro/2019/05/21/rsnHnp5L.gz cve bitcoin-address	71	

Showing 1 to 3 of 3 entries

MISP

MISP Taxonomies

- **Tagging** is a simple way to attach a classification to an event or anattribute.
- **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 [5].
- Can be easily cherry-picked or extended

[5] 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
- **fpf**[6]: Evaluate the degree of identifiability of personal data and the types of pseudonymous data, de-identified data and anonymous data.

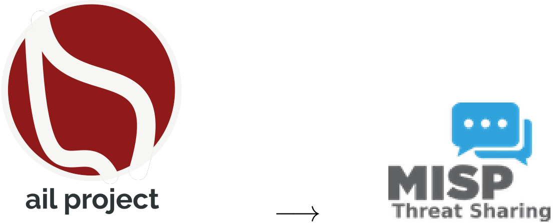
[6] Future of Privacy Forum

Taxonomies useful in AIL

- **tor**: Describe Tor network infrastructure.
- **dark-web**: Criminal motivation on the dark web.
- **copine-scale**[7]: Categorise the severity of images of child sex abuse.

[7] 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

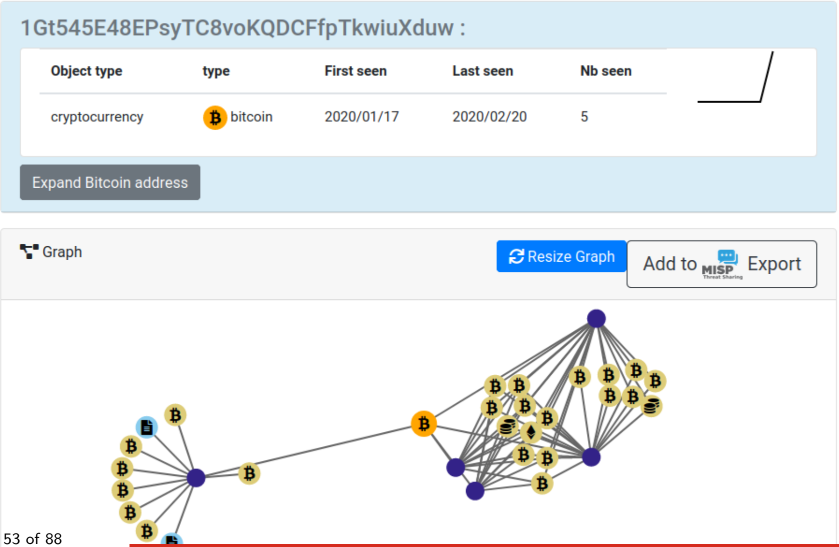
ail project → MISP Threat Sharing

1. Use infoleak taxonomy[8]
2. Add your own tags
3. Export AIL objects to MISP core format
4. Download it or Create a MISP Event[9]

[8] https://www.misp-project.org/taxonomies.html

[9] https://www.misp-standard.org/rfc/misp-standard-core.txt

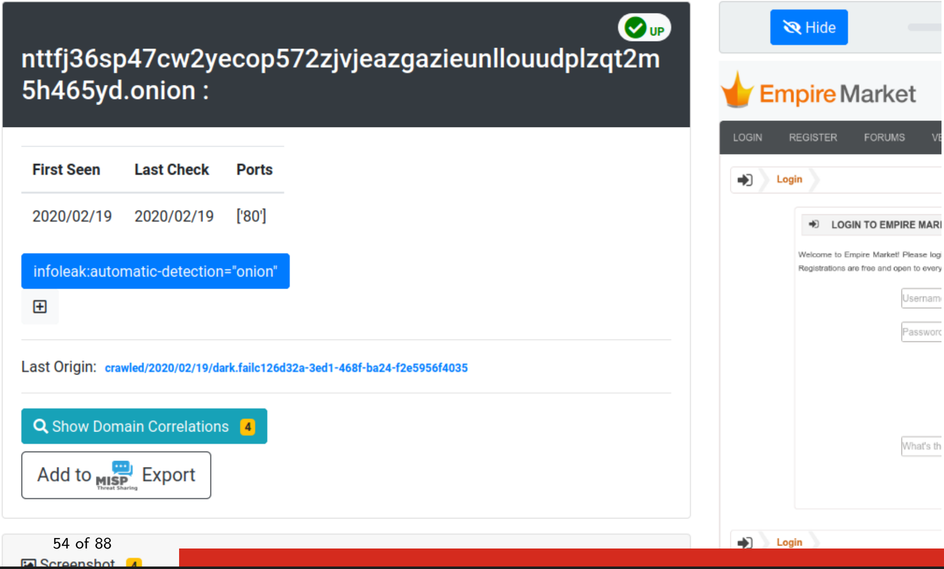
MISP Export



1Gt545E48EPsyTC8voKQDCFfpTkwiuXduw :

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

MISP Export



UP

nttfj36sp47cw2yecop572zjvjeazgazieunllouudplzqt2m5h465yd.onion :

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

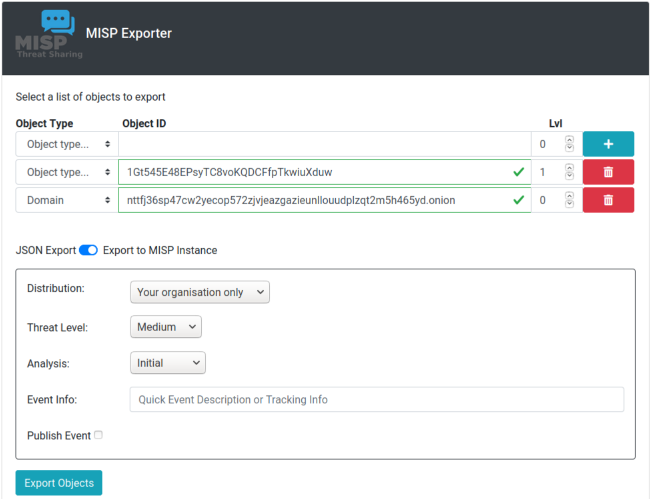
infoleak:automatic-detection="onion"

Last Origin: crawled/2020/02/19/dark.failc126d32a-3ed1-468f-ba24-f2e5956f4035

Welcome to Empire Market! Please logi

Registrations are free and open to every

MISP Export



MISP Exporter

Select a list of objects to export

Object Type	Object ID	Lvl	
Object type...		0	+
Object type...	1Gt545E48EPsyTC8voKQDCFfpTkwiuXduw ✓	1	🗑
Domain	nttfj36sp47cw2yecop572zjvjeazgazieunllouudplzqt2m5h465yd.onion ✓	0	🗑

JSON Export ● Export to MISP Instance

Distribution: Your organisation only

Threat Level: Medium

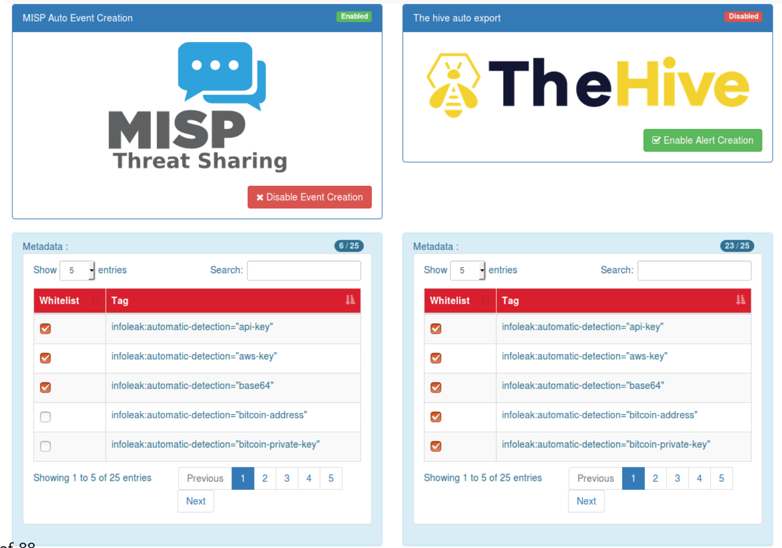
Analysis: Initial

Event Info: Quick Event Description or Tracking Info

Publish Event □

Export Objects

Automatic submission on tags



MISP Auto Event Creation Enabled

✖ Disable Event Creation

Metadata : 6 / 25

Show 5 entries Search:

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

Showing 1 to 5 of 25 entries

Previous 1 2 3 4 5 Next

The hive auto export Disabled

☑ Enable Alert Creation

Metadata : 23 / 25

Show 5 entries Search:

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

Showing 1 to 5 of 25 entries

Previous 1 2 3 4 5 Next

API

AIL exposes a ReST API which can be used to interact with the back-end[10].

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

```

- AIL API is currently covering 60% of the functionality of back-end.

[10] https://github.com/ail-project/ail-framework/blob/master/doc/README.md

Setting up the framework

Setting up AIL-Framework from source or virtual machine

Setting up AIL-Framework from source

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

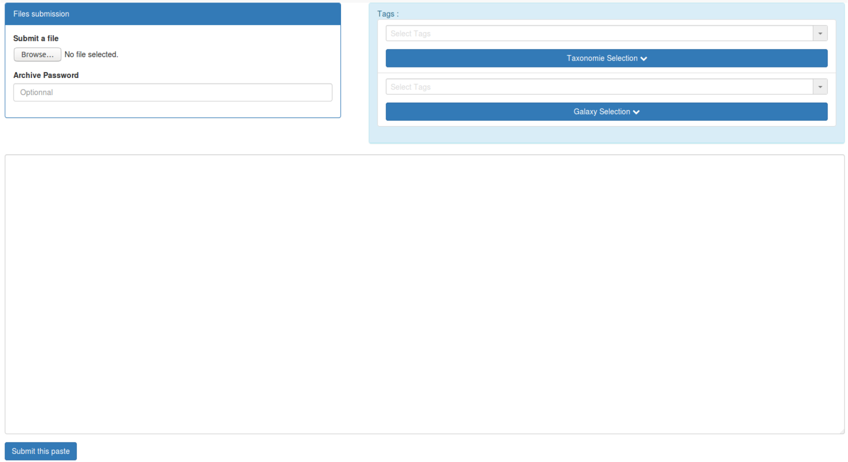
Feeding the framework

Feeding AIL

There are different way to feed AIL with data:

1. Setup *pystemon* and use the custom feeder
   - *pystemon* will collect items for you
2. Use the new JSON Feeder (twitter)
3. Feed your own data using the API or the `import_dir.py` script
4. Feed your own file/text using the UI (`Submit section`)

Via the UI (1)



Files submission

Submit a file

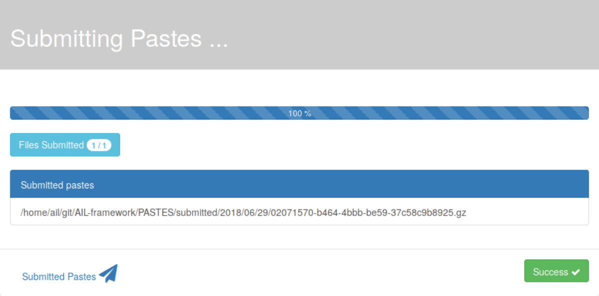
Browse... No file selected.

Archive Password

Optionnal

Tags :

Via the UI (2)



Submitting Pastes ...

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

Feeding AIL with your own data - API

api/v1/import/item

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

Feeding AIL with Twitter posts and associated urls

- AIL - feeder from Twitter[11]
- The AIL-feeder-twitter search in Twitter using Twint (without API), crawls the urls and pushes the results in AIL
- The JSON format format can be extended via meta fields

[11] https://github.com/ail-project/ail-feeder-twitter

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

/!\ requirements:

- 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
  - If you want to feed a large file, better split it in multiple ones

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

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

Starting the framework

Running your own instance from source

Accessing the environment and starting AIL

```

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

Running your own instance using the virtual machine

Login and passwords:

```
# Web interface (default network settings)
    https://127.0.0.1:7000/
# Web interface:
    admin@admin.test
    Password1234
# SSH:
    ail
    Password1234
```

Updating AIL

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

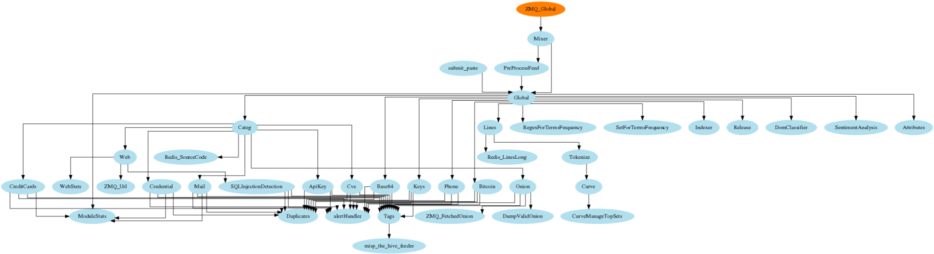
Programming language: Full python3

Databases: Redis and ARDB

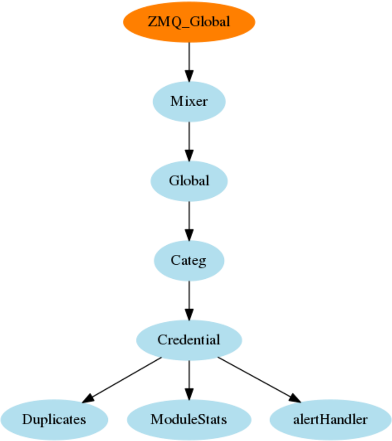
Server: Flask

Data message passing: ZMQ, Redis list and Redis Publisher/Subscriber

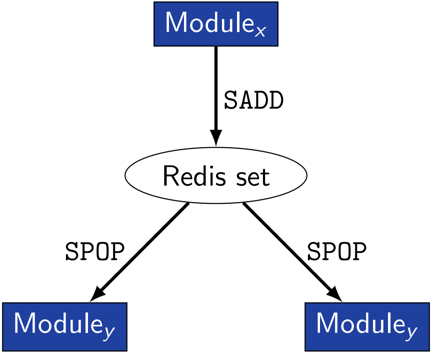
AIL global architecture: Data streaming between module



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



Message consuming



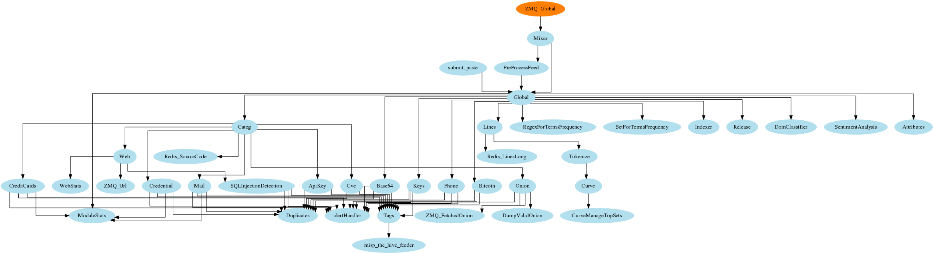
→ No message lost nor double processing

→ 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 `bin/package/modules.cfg` accordingly

Writing your own modules - /bin/template.py

```python
import time
from pubsublogger import publisher
from Helper import Process
if __name__ == '__main__':
    # logger setup
    publisher.port = 6380
    publisher.channel = 'Script'
    # Section name in configs/core.cfg
    config_section = '<section name>'
    # Setup the I/O queues
    p = Process(config_section)
    # Endless loop getting messages from the input queue
    while True:
        # Get one message from the input queue
        message = p.get_from_set()
        if message is None:
            publisher.debug("{} queue is empty, waiting".format(config_section))
            time.sleep(1)
            continue
        # Do something with the message from the queue
        something_has_been_done = do_something(message)

```

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!

く(^.^)ノ

Final words

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

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

Ongoing developments

- New JSON feeders
- Python API wrapper
- Data retention (export/import)
- MISP modules expansion
- auto Classify content by set of terms
  - CE contents
  - DDOS booters
  - ...
- Crawled items
  - duplicate crawled domains
  - tor indexer

Annexes

Managing AIL: Old fashion way

Access the script screen

```
screen -r Script
```

Table: GNU screen shortcuts

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