30/03/2018

Creation of a module in AIL-framework

Discovery of AIL and creation of a game related module

Kowalski Laurent and Kieffer Joris

M2 SSI METZ 2017/2018

SUMMARY

INTRODUCTION

During our last year of master’s Degree in IT Security we had a course about Cyber Threat Intelligence with the professional speaker M. A. Dulaunoy.

We were asked to create a project about a subject or a software in IT Security. During this course we had the chance to discover different kind of software like MISP or AIL-framework. We also discovered other subjects like Malware classifier or how to recover data from a source.

We had an interest in the process of analyzing pastes to find data leaks or potential threats that can happen in the future.

So, we decided to work on AIL-framework. We were told that in these pastes they were a lot of useless data like Game conversations or Game related subjects. With this idea, we decided to create a module to find all these pastes containing Game related words and integer it into AIL-framework.

This report will be divided in 3 parts:

In the beginning we will talk about our discovery of AIL-framework, then we will talk about the module created and finally we will talk about the results of this module.

LOGO OF AIL-FRAMEWORK

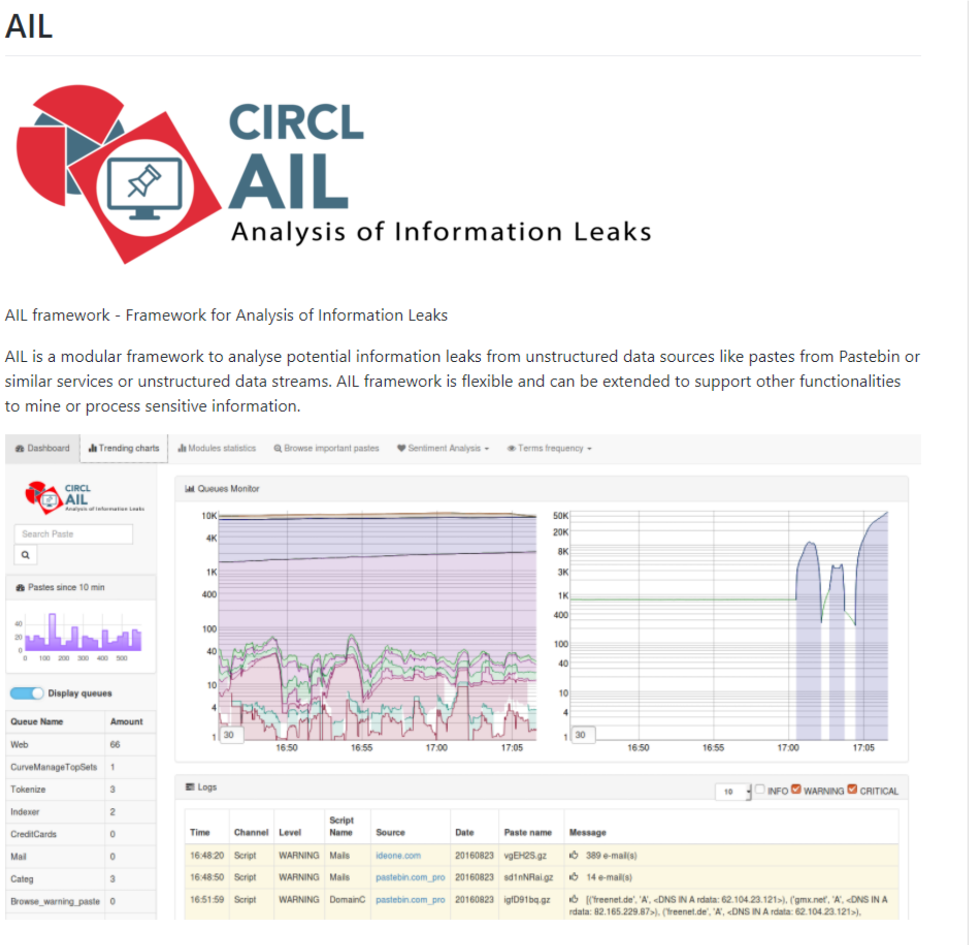
DISCOVERY OF AIL-FRAMEWORK

INTRODUCTION

AIL is a modular framework to analyse potential information leaks from unstructured data sources like pastes from Pastebin or similar services or unstructured data streams. AIL framework is flexible and can be extended to support other functionalities to mine or process sensitive information.

https://github.com/CIRCL/AIL-framework

This framework was created by the CIRCL (Computer Incident Response Center Luxembourg) and you can use it freely and modify it.



AIL

AIL framework - Framework for Analysis of Information Leaks

AIL is a modular framework to analyse potential information leaks from unstructured data sources like pastes from Pastebin or similar services or unstructured data streams. AIL framework is flexible and can be extended to support other functionalities to mine or process sensitive information.

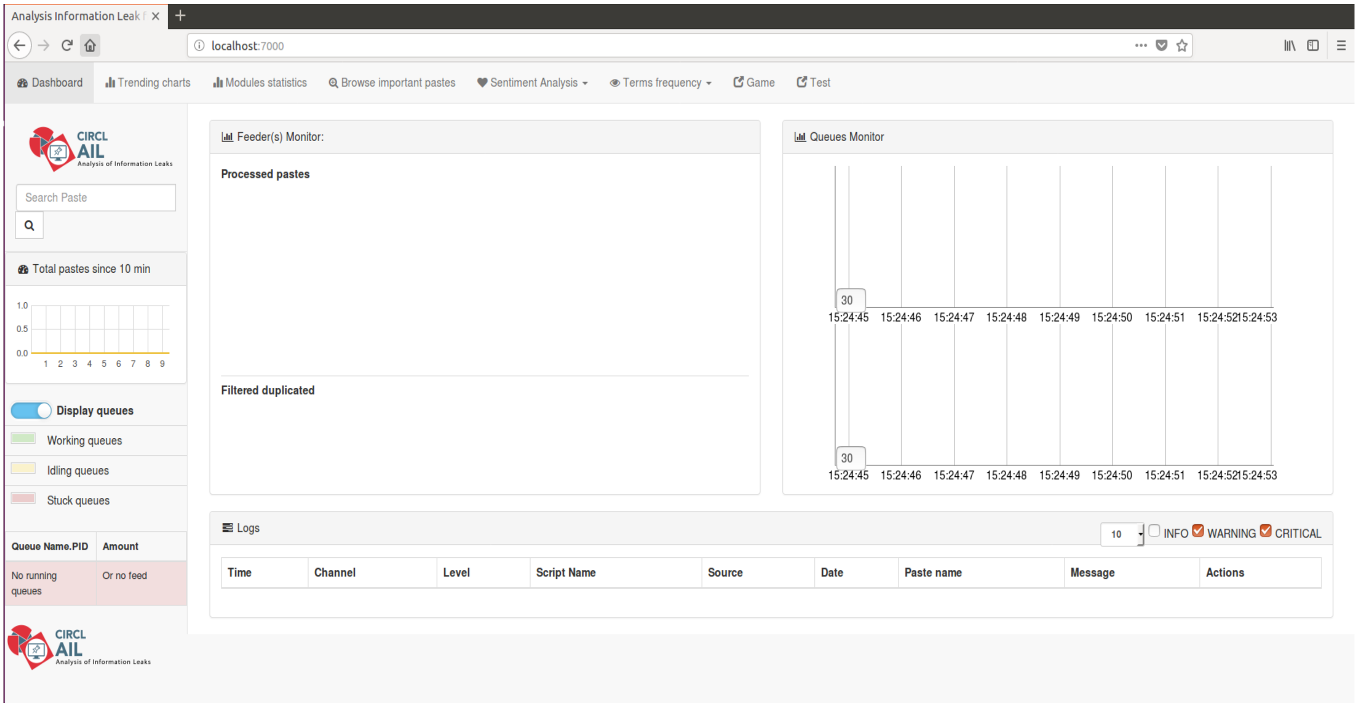
Queue Name	Amount
Web	66
CurveManageTopSets	1
Tokenize	3
Indexer	2
CreditCards	0
Mail	0
Categ	3
Browse_warning_paste	0

Time	Channel	Level	Script Name	Source	Date	Paste name	Message
16:48:20	Script	WARNING	Mails	ideone.com	20160823	vgEH2S.gz	389 e-mail(s)
16:48:50	Script	WARNING	Mails	pastebin.com_pro	20160823	sd1nNRai.gz	14 e-mail(s)
16:51:59	Script	WARNING	DomainC	pastebin.com_pro	20160823	igfD91bq.gz	[('freenet.de', 'A', <DNS IN A rdata: 62.104.23.121>), ('gmx.net', 'A', <DNS IN A rdata: 82.165.229.87>), ('freenet.de', 'A', <DNS IN A rdata: 62.104.23.121>),

FRAMEWORK

FEATURES

- AIL-framework uses modules to analyze different kind of data. You can for example search for credentials or email addresses or even phone numbers.
- It has a web interface allowing you to watch the process and the data found. You can even find graphics about the result of the modules.

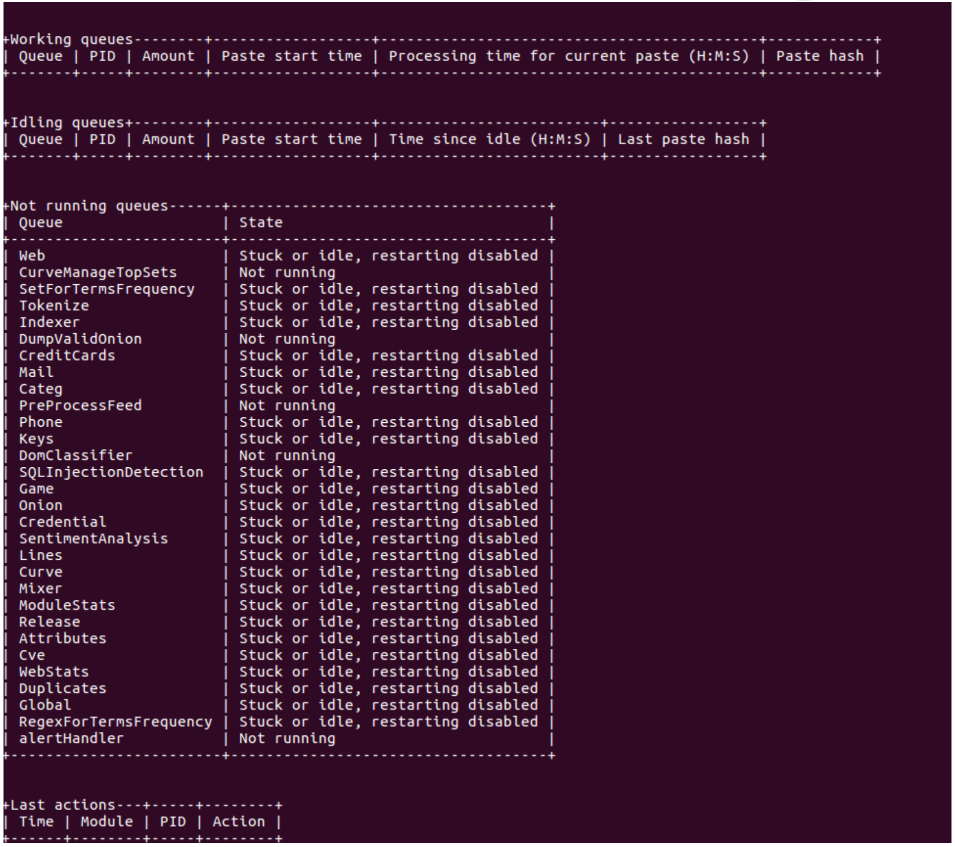


Queue Name.PID	Amount
No running queues	Or no feed

Time	Channel	Level	Script Name	Source	Date	Paste name	Message	Actions

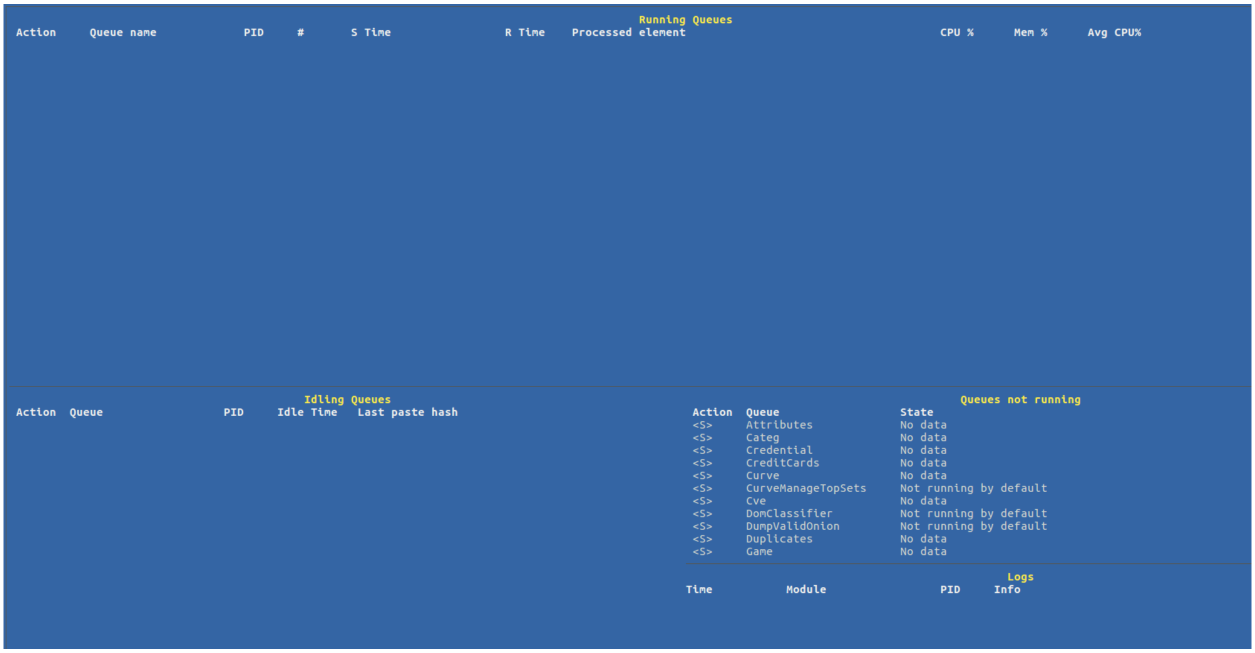
WEB INTERFACE

- This web interface uses the local port 7000.
- On this interface you can see the pastes processed and what the framework found.
- There is a way to watch how the modules are working in 2 different python files: ModuleInformation.py and ModulesInformationV2.py. With these 2 files you can see if a module is working and start it or stop it as you want. It is simple to watch the modules.
- Alerting to MISP to share found leaks within a threat intelligence platform using MISP standard



```
+Working queues--------+-----------------+------------------------------------------+------------+
| Queue | PID | Amount | Paste start time | Processing time for current paste (H:M:S) | Paste hash |
+-------+-----+--------+------------------+------------------------------------------+------------+


+Idling queues+--------+------------------+-------------------------+-----------------+
| Queue | PID | Amount | Paste start time | Time since idle (H:M:S) | Last paste hash |
+-------+-----+--------+------------------+-------------------------+-----------------+


+Not running queues------+-------------------------------------+
| Queue                  | State                               |
+------------------------+-------------------------------------+
| Web                    | Stuck or idle, restarting disabled |
| CurveManageTopSets     | Not running                        |
| SetForTermsFrequency   | Stuck or idle, restarting disabled |
| Tokenize               | Stuck or idle, restarting disabled |
| Indexer                | Stuck or idle, restarting disabled |
| DumpValidOnion         | Not running                        |
| CreditCards            | Stuck or idle, restarting disabled |
| Mail                   | Stuck or idle, restarting disabled |
| Categ                  | Stuck or idle, restarting disabled |
| PreProcessFeed         | Not running                        |
| Phone                  | Stuck or idle, restarting disabled |
| Keys                   | Stuck or idle, restarting disabled |
| DomClassifier          | Not running                        |
| SQLInjectionDetection  | Stuck or idle, restarting disabled |
| Game                   | Stuck or idle, restarting disabled |
| Onion                  | Stuck or idle, restarting disabled |
| Credential             | Stuck or idle, restarting disabled |
| SentimentAnalysis      | Stuck or idle, restarting disabled |
| Lines                  | Stuck or idle, restarting disabled |
| Curve                  | Stuck or idle, restarting disabled |
| Mixer                  | Stuck or idle, restarting disabled |
| ModuleStats            | Stuck or idle, restarting disabled |
| Release                | Stuck or idle, restarting disabled |
| Attributes             | Stuck or idle, restarting disabled |
| Cve                    | Stuck or idle, restarting disabled |
| WebStats               | Stuck or idle, restarting disabled |
| Duplicates             | Stuck or idle, restarting disabled |
| Global                 | Stuck or idle, restarting disabled |
| RegexForTermsFrequency | Stuck or idle, restarting disabled |
| alertHandler           | Not running                        |
+------------------------+-------------------------------------+


+Last actions---+-----+--------+
| Time | Module | PID | Action |
+------+--------+-----+--------+
```

MODULEINFORMATION.PY



```
                                                        Running Queues
Action    Queue name          PID    #      S Time              R Time    Processed element                          CPU %     Mem %     Avg CPU%




                                       Idling Queues                                                          Queues not running
Action  Queue                PID    Idle Time   Last paste hash            Action  Queue                State
                                                                           <S>     Attributes           No data
                                                                           <S>     Categ                No data
                                                                           <S>     Credential           No data
                                                                           <S>     CreditCards          No data
                                                                           <S>     Curve                No data
                                                                           <S>     CurveManageTopSets   Not running by default
                                                                           <S>     Cve                  No data
                                                                           <S>     DomClassifier        Not running by default
                                                                           <S>     DumpValidOnion       Not running by default
                                                                           <S>     Duplicates           No data
                                                                           <S>     Game                 No data
                                                                                                          Logs
                                                                          Time          Module             PID    Info
```

MODULESINFORMATIONSV2.PY

- The second interface is easier to start or stop a module.
- You can use different kind of sources for your analysis. You can use live feed from a source you had access. Or you can upload your own files using the import file. For this project we will use the import file with a dataset we got containing the pastes of December 2017.
- Every action or information found by the modules is logged into files in AIL-framework/logs
- You can create your own modules using the template they give you. And that’s what we will do now.

CREATION OF THE MODULE

To create a module about game words detection in AIL we needed to work on 2 separate things:

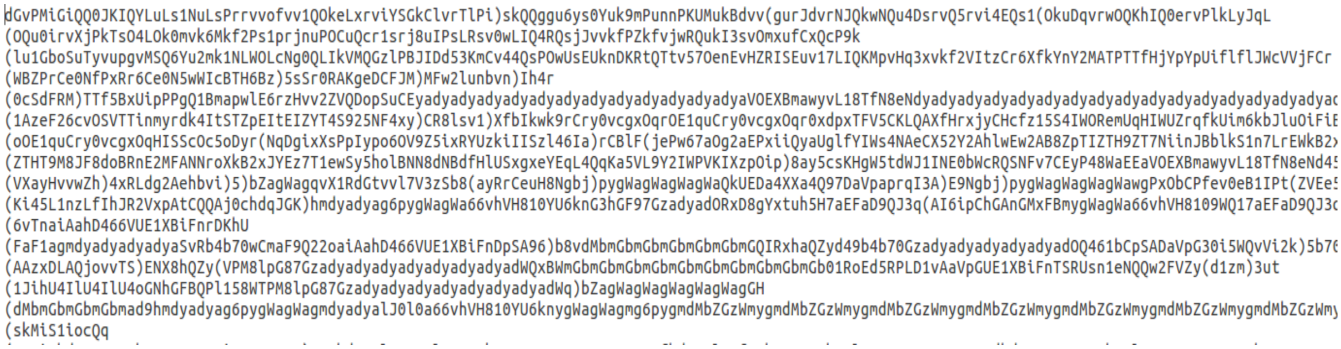
- The list of words used for the detection and the false positives that can result from a bad list
- The module and it's integration in the framework

CORPUS OF GAME RELATED WORD/ EXPRESSIONS

First, we needed to see if a list of this terms was already available online, so we found this page which contains a lot of game related terms:

https://fr.wikipedia.org/wiki/Lexique_du_jeu_vid%C3%A9o

Many of these words are Anglicism so we can use them for our project. So, we started with all the words on the page. But something came up, we got too much files detected and many of them were not game related. A lot of files were detected because some expression in gaming are 2 letters long so if a file is a very big file with random text or code for a malware, it can be detected as a game related file. We don't want this to happen.



```
dGvPMiGiQQ0JKIQYLuLs1NuLsPrrvvofvv1QOkeLxrviYSGkClvrTlPi)skQQggu6ys0Yuk9mPunnPKUMukBdvv(gurJdvrNJQkwNQu4DsrvQ5rvi4EQs1(OkuDqvrwOQKhIQ0ervPlkLyJqL
(OQu0irvXjPkTsO4LOk0mvk6Mkf2Ps1prjnuPOCuQcr1srj8uIPsLRsv0wLIQ4RQsjJvvkfPZkfvjwRQukI3svOmxufCxQcP9k
(lu1GboSuTyvupgvMSQ6Yu2mk1NLWOLcNg0QLIkVMQGzlPBJIDd53KmCv44QsPOwUsEUknDKRtQTtv57OenEvHZRISEuv17LIQKMpvHq3xvkf2VItzCr6XfkYnY2MATPTTfHjYpYpUiflflJWcVVjFCr
(WBZPrCe0NfPxRr6Ce0N5wWIcBTH6Bz)5sSr0RAKgeDCFJM)MFw2lunbvn)Ih4r
(0cSdFRM)TTf5BxUipPPgQ1BmapwlE6rzHvv2ZVQDopSuCEyadyadyadyadyadyadyadyadyadyadyadyaVOEXBmawyvL18TfN8eNdyadyadyadyadyadyadyadyadyadyadyadyadyadyadyac
(1AzeF26cvOSVTTinmyrdk4ItSTZpEItEIZYT4S925NF4xy)CR8lsv1)XfbIkwk9rCry0vcgxOqrOE1quCry0vcgxOqr0xdpxTFV5CKLQAXfHrxjyCHcfz15S4IWORemUqHIWUZrqfkUim6kbJluOiFiE
(oOE1quCry0vcgxOqHISScOc5oDyr(NqDgixXsPpIypo6OV9Z5ixRYUzkiIISzl46Ia)rCBlF(jePw67aOg2aEPxiiQyaUglfYIWs4NAeCX52Y2AhlwEw2AB8ZpTIZTH9ZT7NiinJBblkS1n7LrEWkB2x
(ZTHT9M8JF8doBRnE2MFANNroXkB2xJYEz7T1ewSy5holBNN8dNBdfHlUSxgxeYEqL4QqKa5VL9Y2IWPVKIXzpOip)8ay5csKHgW5tdWJ1INE0bWcRQSNFv7CEyP48WaEEaVOEXBmawyvL18TfN8eNd45
(VXayHvvwZh)4xRLdg2Aehbvi)5)bZagWagqvX1RdGtvvl7V3zSb8(ayRrCeuH8Ngbj)pygWagWagWagWaQkUEDa4XXa4Q97DaVpaprqI3A)E9Ngbj)pygWagWagWagWawgPxObCPfev0eB1IPt(ZVEe5
(Ki45L1nzLfIhJR2VxpAtCQQAj0chdqJGK)hmdyadyag6pygWagWa66vhVH810YU6knG3hGF97GzadyadORxD8gYxtuh5H7aEFaD9QJ3q(AI6ipChGAnGMxFBmygWagWa66vhVH8109WQ17aEFaD9QJ3c
(6vTnaiAahD466VUE1XBiFnrDKhU
(FaF1agmdyadyadyadyaSvRb4b70wCmaF9Q22oaiAahD466VUE1XBiFnDpSA96)b8vdMbmGbmGbmGbmGbmGbmGQIRxhaQZyd49b4b70GzadyadyadyadyadOQ461bCpSADaVpG30i5WQvVi2k)5b76
(AAzxDLAQjovvTS)ENX8hQZy(VPM8lpG87GzadyadyadyadyadyadyadyadWQxBWmGbmGbmGbmGbmGbmGbmGbmGbmGbmGb01RoEd5RPLD1vAaVpGUE1XBiFnTSRUsn1eNQQw2FVZy(d1zm)3ut
(1JihU4IlU4IlU4oGNhGFBQPl158WTPM8lpG87GzadyadyadyadyadyadyadyadWq)bZagWagWagWagWagWagGH
(dMbmGbmGbmGbmad9hmdyadyag6pygWagWagmdyadyalJ0l0a66vhVH810YU6knygWagWagmg6pygmdMbZGzWmygmdMbZGzWmygmdMbZGzWmygmdMbZGzWmygmdMbZGzWmygmdMbZGzWmygmdMbZGzWmy
(skMiS1iocQq
```

EXAMPLE OF FILE DETECTED WITHOUT CHECKING THE WORDS FIRST

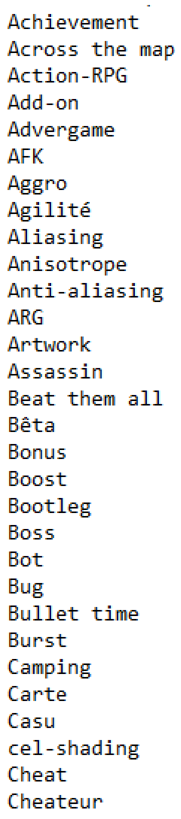
So, a solution for our problem was to erase all words with a length inferior to 3 characters in order to prevent this kind of problems.

Another problem was the following: How many different words related to games in a file to make it a game related conversation?

We decided to put it at 5 occurrences of different game words because the goal is to find real data about games and not a false positive with 20 occurrences of the same word

We also thought which kind of users/gamers will use pastes to give information and after some first analysis we noticed that most of the files were configuration file for games and MMO related conversation, so we added a list of words from the world of MMORPG and for the configuration file they were already in the list.

When we are writing this report, the list is made of about 250+ words/expressions/games



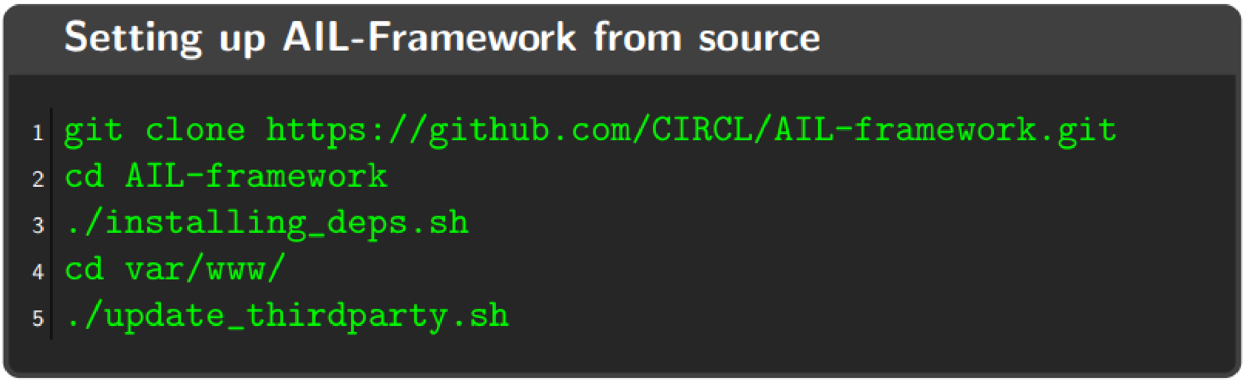
```
Achievement
Across the map
Action-RPG
Add-on
Advergame
AFK
Aggro
Agilité
Aliasing
Anisotrope
Anti-aliasing
ARG
Artwork
Assassin
Beat them all
Bêta
Bonus
Boost
Bootleg
Boss
Bot
Bug
Bullet time
Burst
Camping
Carte
Casu
cel-shading
Cheat
Cheateur
```

EXAMPLE OF THE WORDS YOU CAN FIND FOR THIS MODULE

All these words can be found in the corpus.txt file in the github, this list can be modified to be more accurate or for another subject other than games.

THE MODULE AND ITS INTEGRATION IN THE AIL FRAMEWORK

We installed AIL framework on linux x64 virtual machine using the following tutorial:

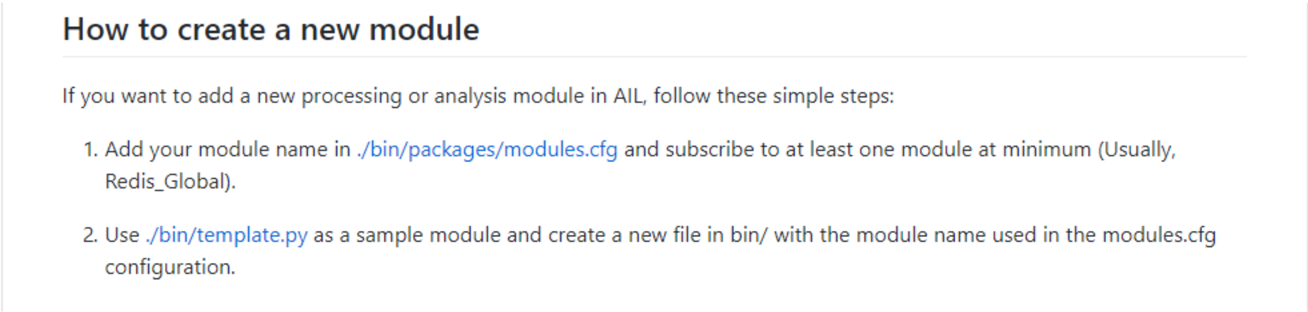


Setting up AIL-Framework from source

```
git clone https://github.com/CIRCL/AIL-framework.git
cd AIL-framework
./installing_deps.sh
cd var/www/
./update_thirdparty.sh
```

TUTORIAL TO INSTALL AIL

The first step to create a module can be found on the howto.md file from the AIL-framework github page:



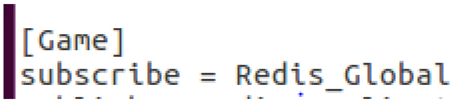
How to create a new module

If you want to add a new processing or analysis module in AIL, follow these simple steps:

1. Add your module name in ./bin/packages/modules.cfg and subscribe to at least one module at minimum (Usually, Redis_Global).
2. Use ./bin/template.py as a sample module and create a new file in bin/ with the module name used in the modules.cfg configuration.

HOW TO CREATE A MODULE

So, we added these lines in the modules.cfg file:

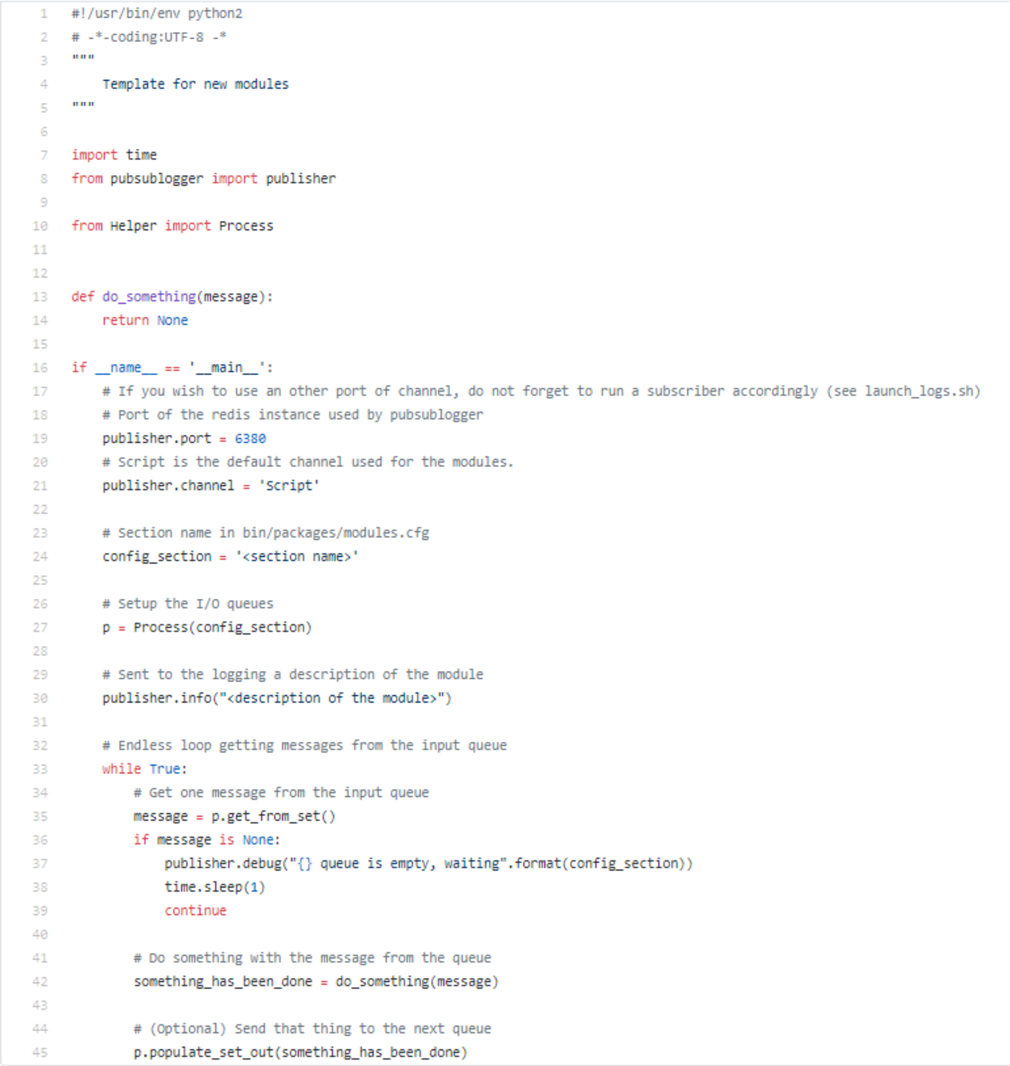


```
[Game]
subscribe = Redis_Global
```

MODULES.CFG

So, our module will be called Game and will use Redis_Global. For now, the module will not send anything to other modules but the idea of using the Game module as a filter before other modules is an idea we thought about.

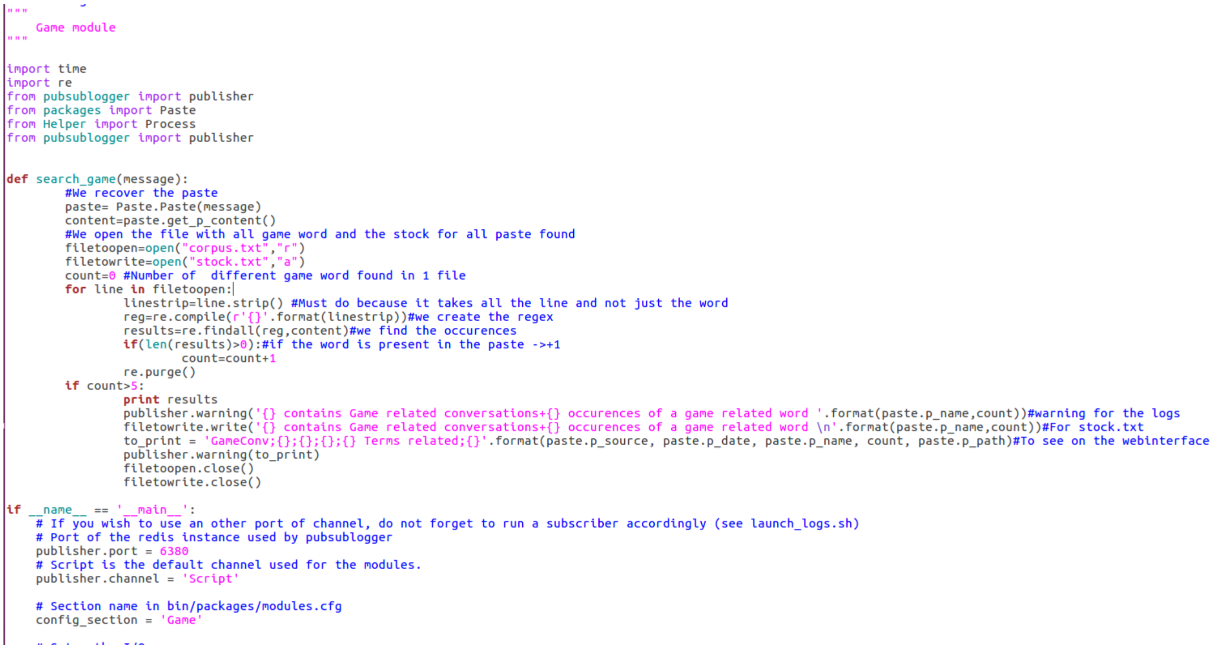
For the module file, we will use the template and just modify what we need to make it work:



```
#!/usr/bin/env python2
# -*-coding:UTF-8 -*
"""
    Template for new modules
"""

import time
from pubsublogger import publisher

from Helper import Process


def do_something(message):
    return None

if __name__ == '__main__':
    # If you wish to use an other port of channel, do not forget to run a subscriber accordingly (see launch_logs.sh)
    # 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)

        # (Optional) Send that thing to the next queue
        p.populate_set_out(something_has_been_done)
```

TEMPLATE

To make this module we will use regex expressions with every word in the corpus.txt file. And in order to stock the results we copy the file found and the number of occurrences in stock.txt

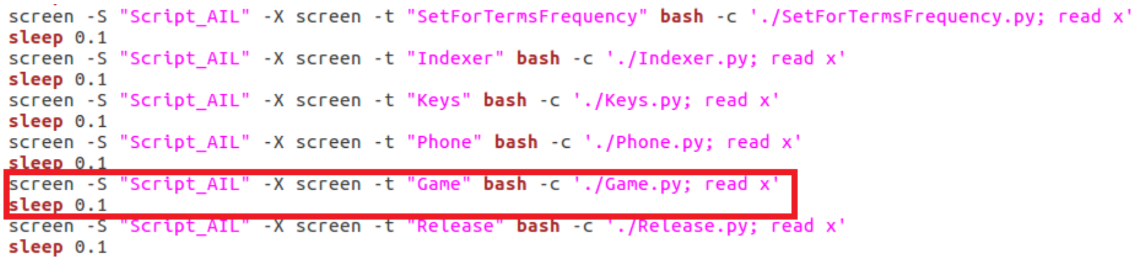


```
"""
    Game module
"""

import time
import re
from pubsublogger import publisher
from packages import Paste
from Helper import Process
from pubsublogger import publisher


def search_game(message):
        #We recover the paste
        paste= Paste.Paste(message)
        content=paste.get_p_content()
        #We open the file with all game word and the stock for all paste found
        filetoopen=open("corpus.txt","r")
        filetowrite=open("stock.txt","a")
        count=0 #Number of  different game word found in 1 file
        for line in filetoopen:
                linestrip=line.strip() #Must do because it takes all the line and not just the word
                reg=re.compile(r'{}'.format(linestrip))#we create the regex
                results=re.findall(reg,content)#we find the occurences
                if(len(results)>0):#if the word is present in the paste ->+1
                        count=count+1
                re.purge()
        if count>5:
                print results
                publisher.warning('{} contains Game related conversations+{} occurences of a game related word '.format(paste.p_name,count))#warning for the logs
                filetowrite.write('{} contains Game related conversations+{} occurences of a game related word \n'.format(paste.p_name,count))#For stock.txt
                to_print = 'GameConv;{};{};{};{} Terms related;{}'.format(paste.p_source, paste.p_date, paste.p_name, count, paste.p_path)#To see on the webinterface
                publisher.warning(to_print)
                filetoopen.close()
                filetowrite.close()

if __name__ == '__main__':
    # If you wish to use an other port of channel, do not forget to run a subscriber accordingly (see launch_logs.sh)
    # 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 = 'Game'
```

FINAL CODE OF GAME.PY

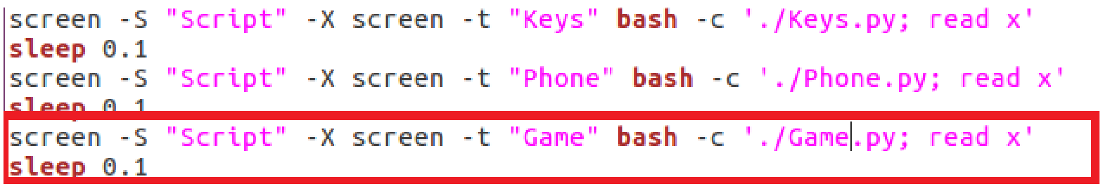
To use it now, it wasn't enough to just modify these 2 files. We needed to modify another file to make it work with AIL.

First, we added our module in both launch files for AIL-framework: LAUNCH.sh and launch_scripts.sh:



```
screen -S "Script_AIL" -X screen -t "SetForTermsFrequency" bash -c './SetForTermsFrequency.py; read x'
sleep 0.1
screen -S "Script_AIL" -X screen -t "Indexer" bash -c './Indexer.py; read x'
sleep 0.1
screen -S "Script_AIL" -X screen -t "Keys" bash -c './Keys.py; read x'
sleep 0.1
screen -S "Script_AIL" -X screen -t "Phone" bash -c './Phone.py; read x'
sleep 0.1
screen -S "Script_AIL" -X screen -t "Game" bash -c './Game.py; read x'
sleep 0.1
screen -S "Script_AIL" -X screen -t "Release" bash -c './Release.py; read x'
sleep 0.1
```

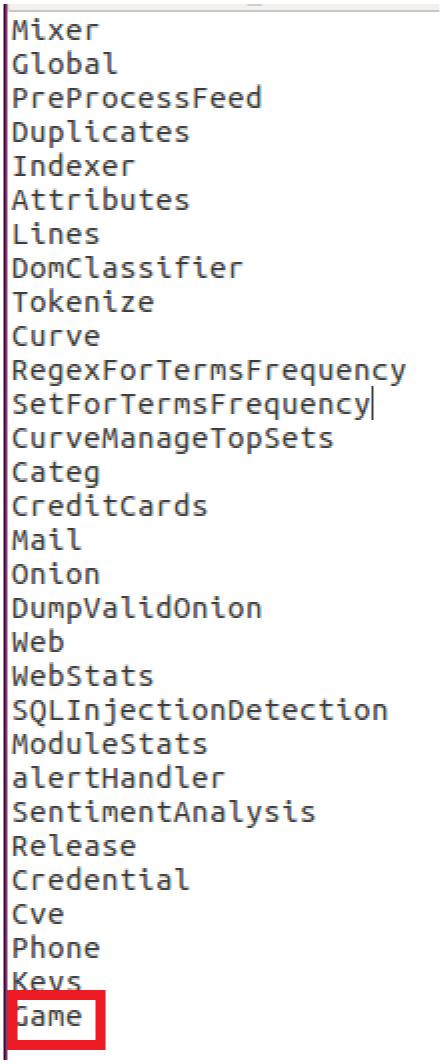
LAUNCH.SH



```
screen -S "Script" -X screen -t "Keys" bash -c './Keys.py; read x'
sleep 0.1
screen -S "Script" -X screen -t "Phone" bash -c './Phone.py; read x'
sleep 0.1
screen -S "Script" -X screen -t "Game" bash -c './Game.py; read x'
sleep 0.1
```

LAUNCH_SCRIPTS.SH

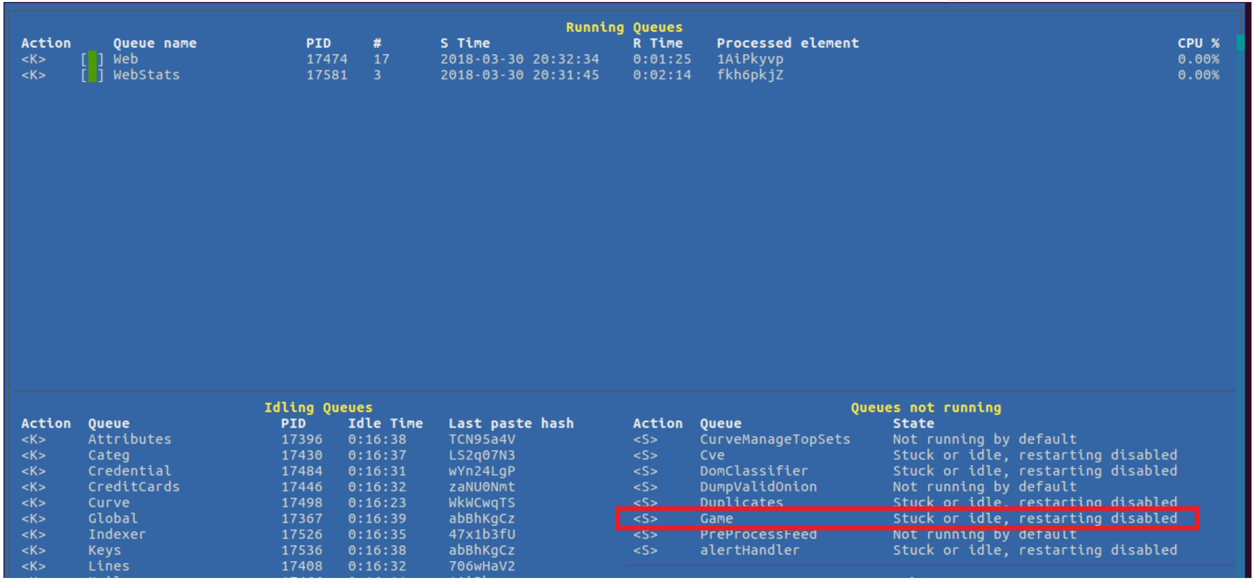
And finally, we needed to add our module into the doc to see it with the appropriate tools (ModuleInformation.py…)



```
Mixer
Global
PreProcessFeed
Duplicates
Indexer
Attributes
Lines
DomClassifier
Tokenize
Curve
RegexForTermsFrequency
SetForTermsFrequency
CurveManageTopSets
Categ
CreditCards
Mail
Onion
DumpValidOnion
Web
WebStats
SQLInjectionDetection
ModuleStats
alertHandler
SentimentAnalysis
Release
Credential
Cve
Phone
Keys
Game
```

FICHIER ALL_MODULES.TXT

Results: we can see our module in the tools

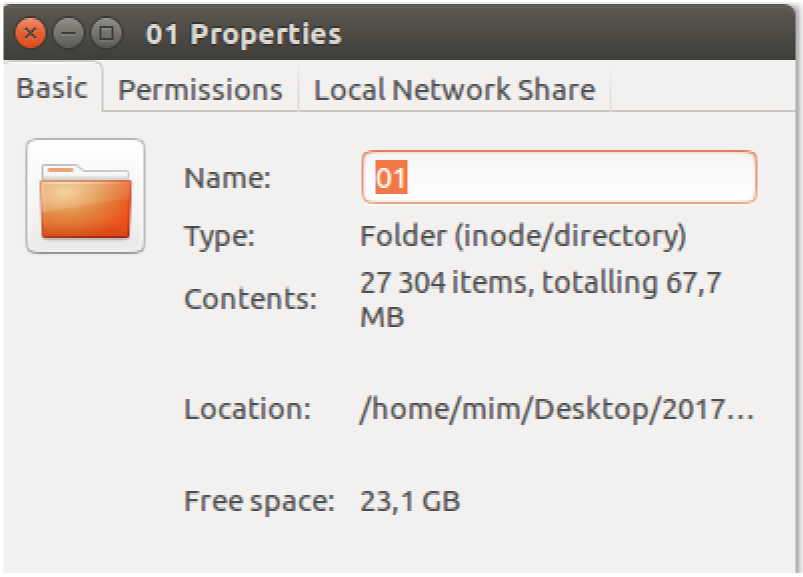


```
                                                                    Running Queues
Action      Queue name              PID      #        S Time                 R Time      Processed element                                          CPU %
<K>      [█] Web                    17474    17       2018-03-30 20:32:34    0:01:25     1AiPkyvp                                                   0.00%
<K>      [█] WebStats               17581    3        2018-03-30 20:31:45    0:02:14     fkh6pkjZ                                                   0.00%

                              Idling Queues                                                           Queues not running
Action   Queue                  PID     Idle Time   Last paste hash      Action   Queue                   State
<K>      Attributes             17396   0:16:38     TCN95a4V             <S>      CurveManageTopSets      Not running by default
<K>      Categ                  17430   0:16:37     LS2q07N3             <S>      Cve                     Stuck or idle, restarting disabled
<K>      Credential             17484   0:16:31     wYn24LgP             <S>      DomClassifier           Stuck or idle, restarting disabled
<K>      CreditCards            17446   0:16:32     zaNU0Nmt             <S>      DumpValidOnion          Not running by default
<K>      Curve                  17498   0:16:23     WkWCwqTS             <S>      Duplicates              Stuck or idle, restarting disabled
<K>      Global                 17367   0:16:39     abBhKgCz             <S>      Game                    Stuck or idle, restarting disabled
<K>      Indexer                17526   0:16:35     47x1b3fU             <S>      PreProcessFeed          Not running by default
<K>      Keys                   17536   0:16:38     abBhKgCz             <S>      alertHandler            Stuck or idle, restarting disabled
<K>      Lines                  17408   0:16:32     706wHaV2
```

OUR MODULE WITH MODULESINFORMATIONV2.PY

RESULTS

To test our module, we had some pastes from December 2017. So, we will test this on the paste of 1 day



Name: 01

Type: Folder (inode/directory)

Contents: 27 304 items, totalling 67,7 MB

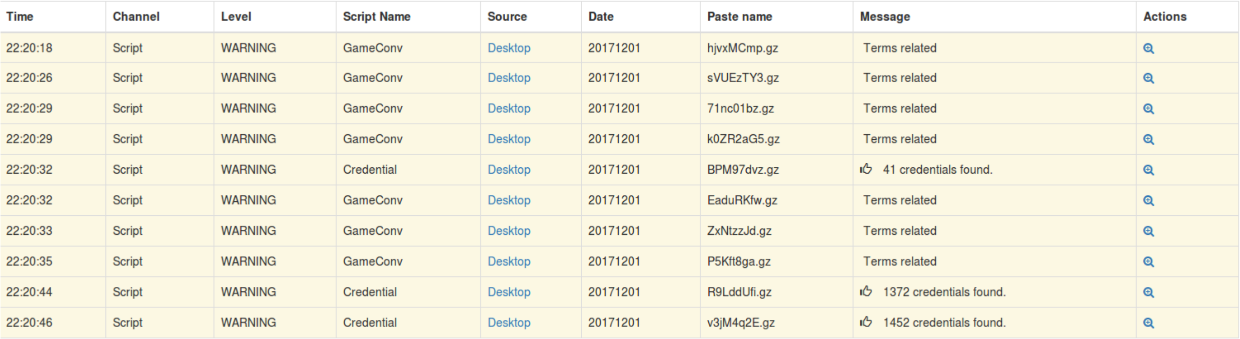
Location: /home/mim/Desktop/2017…

Free space: 23,1 GB

NUMBER OF FILES FOR 1 DAY WORTH OF PASTES

And now we will see how many pastes are related to games, and we will see some of these files.

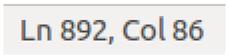
After 1 hour, we decided to stop the analysis. AIL-framework analyzed around 3 000 pastes



Time	Channel	Level	Script Name	Source	Date	Paste name	Message	Actions
22:20:18	Script	WARNING	GameConv	Desktop	20171201	hjvxMCmp.gz	Terms related	🔍
22:20:26	Script	WARNING	GameConv	Desktop	20171201	sVUEzTY3.gz	Terms related	🔍
22:20:29	Script	WARNING	GameConv	Desktop	20171201	71nc01bz.gz	Terms related	🔍
22:20:29	Script	WARNING	GameConv	Desktop	20171201	k0ZR2aG5.gz	Terms related	🔍
22:20:32	Script	WARNING	Credential	Desktop	20171201	BPM97dvz.gz	👍 41 credentials found.	🔍
22:20:32	Script	WARNING	GameConv	Desktop	20171201	EaduRKfw.gz	Terms related	🔍
22:20:33	Script	WARNING	GameConv	Desktop	20171201	ZxNtzzJd.gz	Terms related	🔍
22:20:35	Script	WARNING	GameConv	Desktop	20171201	P5Kft8ga.gz	Terms related	🔍
22:20:44	Script	WARNING	Credential	Desktop	20171201	R9LddUfi.gz	👍 1372 credentials found.	🔍
22:20:46	Script	WARNING	Credential	Desktop	20171201	v3jM4q2E.gz	👍 1452 credentials found.	🔍

RESULTS

Our module found around 892 pastes with game related words in it

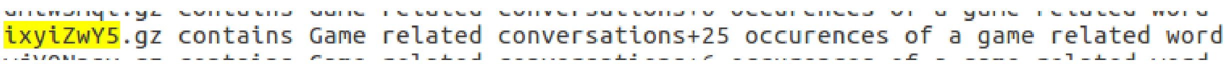


Ln 892, Col 86

NUMBER OF LINES IN STOCK.TXT AFTER THE ANALYSIS

Let’s see some of the pastes found:

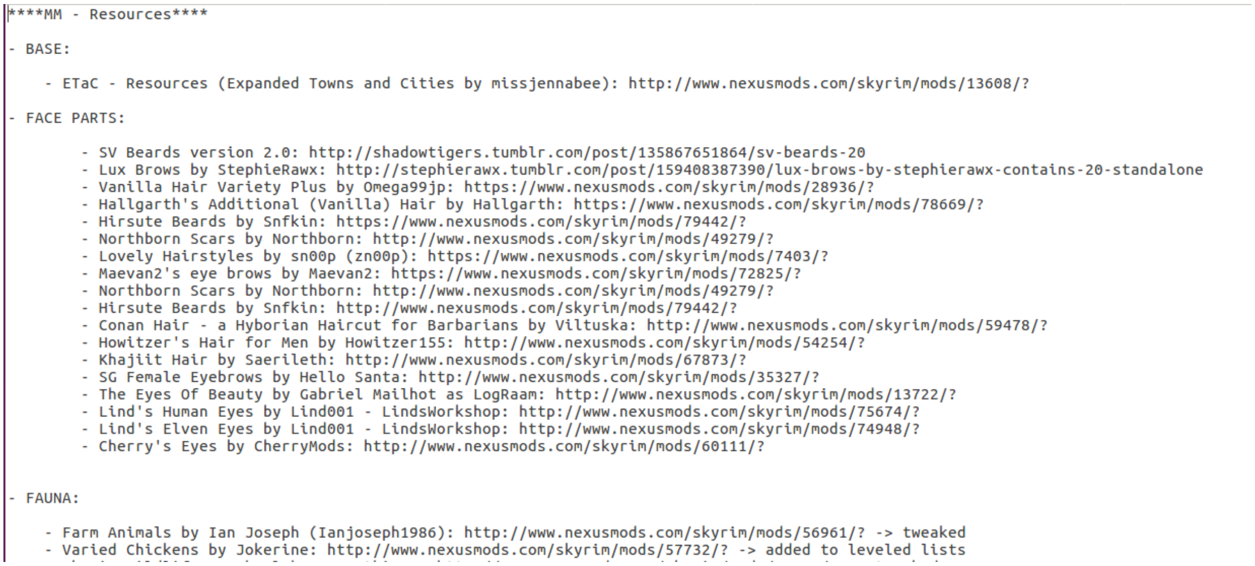
First file:



```
ixyiZwY5.gz contains Game related conversations+25 occurences of a game related word
```

FIRST FILE RANDOMLY CHOSEN

The file:

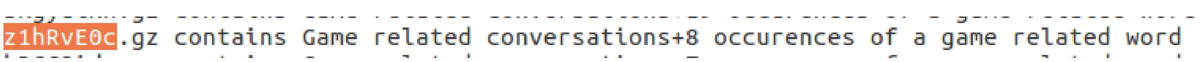


```
****MM - Resources****

- BASE:

    - ETaC - Resources (Expanded Towns and Cities by missjennabee): http://www.nexusmods.com/skyrim/mods/13608/?

- FACE PARTS:

        - SV Beards version 2.0: http://shadowtigers.tumblr.com/post/135867651864/sv-beards-20
        - Lux Brows by StephieRawx: http://stephierawx.tumblr.com/post/159408387390/lux-brows-by-stephierawx-contains-20-standalone
        - Vanilla Hair Variety Plus by Omega99jp: https://www.nexusmods.com/skyrim/mods/28936/?
        - Hallgarth's Additional (Vanilla) Hair by Hallgarth: https://www.nexusmods.com/skyrim/mods/78669/?
        - Hirsute Beards by Snfkin: https://www.nexusmods.com/skyrim/mods/79442/?
        - Northborn Scars by Northborn: http://www.nexusmods.com/skyrim/mods/49279/?
        - Lovely Hairstyles by sn00p (zn00p): https://www.nexusmods.com/skyrim/mods/7403/?
        - Maevan2's eye brows by Maevan2: https://www.nexusmods.com/skyrim/mods/72825/?
        - Northborn Scars by Northborn: http://www.nexusmods.com/skyrim/mods/49279/?
        - Hirsute Beards by Snfkin: http://www.nexusmods.com/skyrim/mods/79442/?
        - Conan Hair - a Hyborian Haircut for Barbarians by Viltuska: http://www.nexusmods.com/skyrim/mods/59478/?
        - Howitzer's Hair for Men by Howitzer155: http://www.nexusmods.com/skyrim/mods/54254/?
        - Khajiit Hair by Saerileth: http://www.nexusmods.com/skyrim/mods/67873/?
        - SG Female Eyebrows by Hello Santa: http://www.nexusmods.com/skyrim/mods/35327/?
        - The Eyes Of Beauty by Gabriel Mailhot as LogRaam: http://www.nexusmods.com/skyrim/mods/13722/?
        - Lind's Human Eyes by Lind001 - LindsWorkshop: http://www.nexusmods.com/skyrim/mods/75674/?
        - Lind's Elven Eyes by Lind001 - LindsWorkshop: http://www.nexusmods.com/skyrim/mods/74948/?
        - Cherry's Eyes by CherryMods: http://www.nexusmods.com/skyrim/mods/60111/?


- FAUNA:

    - Farm Animals by Ian Joseph (Ianjoseph1986): http://www.nexusmods.com/skyrim/mods/56961/? -> tweaked
    - Varied Chickens by Jokerine: http://www.nexusmods.com/skyrim/mods/57732/? -> added to leveled lists
```

IXYIZWY5.TXT

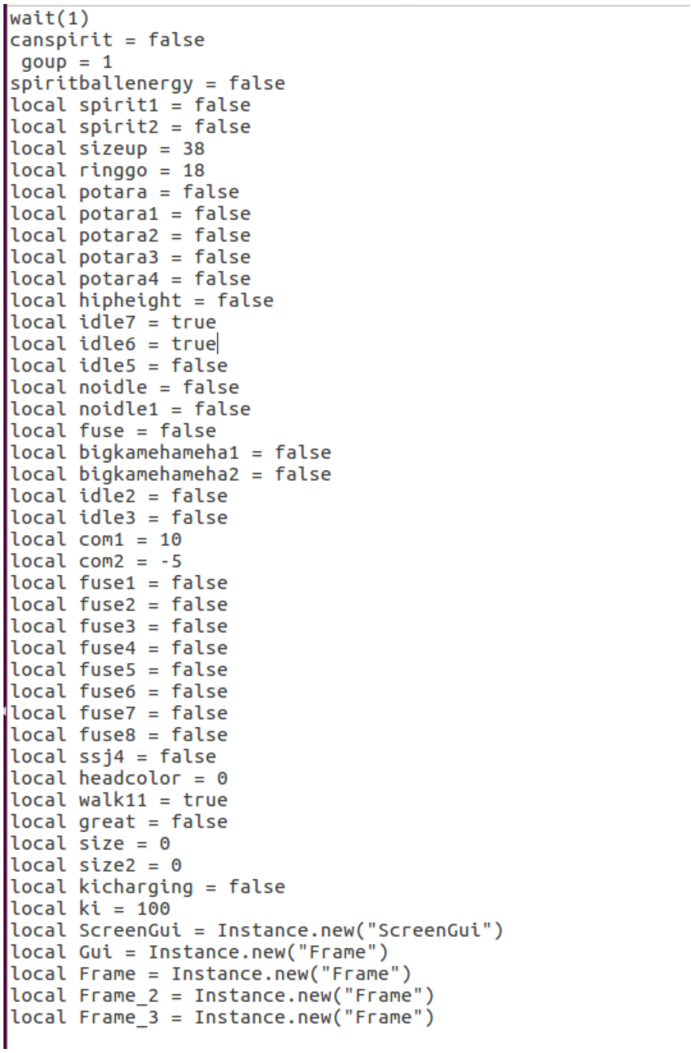
As we can see the file is a simple list of mods for skyrim. So, the module worked and found a game related file.

Another file:



```
z1hRvE0c.gz contains Game related conversations+8 occurences of a game related word
```

SECOND FILE



```
wait(1)
canspirit = false
 goup = 1
spiritballenergy = false
local spirit1 = false
local spirit2 = false
local sizeup = 38
local ringgo = 18
local potara = false
local potara1 = false
local potara2 = false
local potara3 = false
local potara4 = false
local hipheight = false
local idle7 = true
local idle6 = true
local idle5 = false
local noidle = false
local noidle1 = false
local fuse = false
local bigkamehameha1 = false
local bigkamehameha2 = false
local idle2 = false
local idle3 = false
local com1 = 10
local com2 = -5
local fuse1 = false
local fuse2 = false
local fuse3 = false
local fuse4 = false
local fuse5 = false
local fuse6 = false
local fuse7 = false
local fuse8 = false
local ssj4 = false
local headcolor = 0
local walk11 = true
local great = false
local size = 0
local size2 = 0
local kicharging = false
local ki = 100
local ScreenGui = Instance.new("ScreenGui")
local Gui = Instance.new("Frame")
local Frame = Instance.new("Frame")
local Frame_2 = Instance.new("Frame")
local Frame_3 = Instance.new("Frame")
```

Z1HRVE0C.TXT

We can see that this file is maybe a configuration file for a dragonball game. So, the module worked again.

CONCLUSION

In this project we had the chance to discover a very interesting framework and to see how it works.

We created a module and managed to put it in AIL. We had some issues with this part of the project because the first step to create a module was explained but not the work after that to make it work in AIL.

But we managed to do that, and our module works well but we think it can be more accurate if we change the list or the number of game words required to be detected.

We can even think other uses for this module with other type of subject.