Collecting, monitoring and analyzing unstructured data

CIRCL
Computer Incident
Response Center
Luxembourg

TLP:WHITE

AIL framework - Framework for Analysis of Information Leaks

February 4, 2016

Inhalt

- Leak collecting: Pastebin and similar services
  - What is Pastebin?
  - Why Pastebin
  - History
  - Problems
  - Current implementation
- Monitoring - Analysis Information Leak framework
  - Why is Pystemon not enough?
  - Problems
  - Current implementation
- Analysis

Leak collecting - What is Pastebin?

From Wikipedia[1]:

- Web application to publish texts
- Available for a certain time
- Entry available via a a random key
- Usable anonymously

Statistics:

- Pastebin created in 2002
- active pastes 2002: 1.000.000
- active pastes 2011: 10.000.000
- active pastes 2012: 20.000.000
- active pastes 2015: 65.000.000

[1] https://en.wikipedia.org/wiki/Pastebin

More stats

From our Pystemon instance (2016-01): 3.100.000

Sources	Pastes
pastebin.com	2.100.000
ideone.com	470.000
codepad.org	260.000
gist.github.com	140.000
pastebin.ca	50.000
pastebin.ru	40.000
paste.debian.net	20.000
kickasspastes.com	9.000
pastebin.fr	6.000
slexy.org	5.000
lpaste.net	5.000

Leak collecting - Why Pastebin?

Point of view of the attacker:

- Easy to use
- No problem to store big texts
- No moderation
- No registration
- Possible to use anonimity tools for upload

Leak collecting - Why Pastebin?

Point of view of the analyst:

- Lots of dumps of
  - Databases
  - Credit cards
  - Login informations (passwords, keys, ...)
- Very often data concerning organisations in our consistuency

Leak collecting - history

1. Version 1, based on Xavier Garcias Script of June 2011
   - Probably the first script available publicly[2]
   - fetch-pastebin.py: 163 LoC
   - universal-grep.py: 168 LoC
   - CIRCL: Number of words searched: 6
2. Version 2, based on XMEs pastemon[3] of 2012
   - pastemon.pl: 1367 LoC
   - CIRCL: Number of words searched: 41
3. Version 3[4], based on cvandeplas pystemon[5] of 2013
   - pystemon.py: 900 LoC
   - Easy to extend (30 sources implemented)

[2] `http://www.shellguardians.com/2011/07/monitoring-pastebin-leaks.html`

[3] `https://github.com/xme/pastemon`

[4] `https://github.com/CIRCL/pystemon`

[5] `https://github.com/cvandeplas/pystemon`

Leak collecting - problems

- Aggressive download $\rightarrow$ (temp) blacklist
- Respectful download $\rightarrow$ missing pastes
- Multiple IP-Adresses, Multiproxy
- Unicode
- Multithreading

Leak collecting - Current implementation

- New proxy list every day
- Random query through the proxys
- Analyse of the error message (Socket, Timeout, Proxy, Temporary Ban, Blacklist)
- Test of the reliability (based on the amount of errors): removal of the proxy
- If the list of proxy is empty, starts over again
- Saves the pasties in different directories for each service (cdv.lt, codepad.org, gist.github.com, nopaste.me, pastebin.com, pastesite.com, pastie.org, slexy.org, snipt.net)
- Search for pasties based on words
- Sends a mail if it matches

Leak collecting - Current implementation

```
[I] Proxy status: 8 proxies left in memory
[F] Proxy 1.1.2.7:8081 fail count: 3/3
[F] Removing proxy 1.1.2.7:8081 from proxy list because of too many errors.
[I] Proxy status: 7 proxies left in memory
[-] Failed to download the page because of proxy error http://codepad.org/NHj5QagP/raw.txt
[R] Retry 1/100 for http://codepad.org/NHj5QagP/raw.txt
[+] Checking for new pasties from slexy.org. Next download scheduled in 27 seconds
[+] Checking for new pasties from pastebin.com. Next download scheduled in 32 seconds
[+] Found 6 new pasties for site pastebin.com. There are now 6 pasties to be downloaded.
[+] Checking for new pasties from codepad.org. Next download scheduled in 24 seconds
[+] Found 10 new pasties for site codepad.org. There are now 4 pasties to be downloaded.
[+] Checking for new pasties from pastie.org. Next download scheduled in 18 seconds
[+] Checking for new pasties from pastesite.com. Next download scheduled in 22 seconds
[+] Found 2 new pasties for site pastie.org. There are now 2 pasties to be downloaded.
[+] Checking for new pasties from slexy.org. Next download scheduled in 12 seconds
[I] Proxy status: 7 proxies left in memory
[+] Found 12 new pasties for site pastebin.com. There are now 8 pasties to be downloaded.
[A] Found hit for ['Exploit'] in pastie http://pastebin.com/raw.php?i=GgWGJWhb
```

Monitoring - Why is Pystemon not enough?

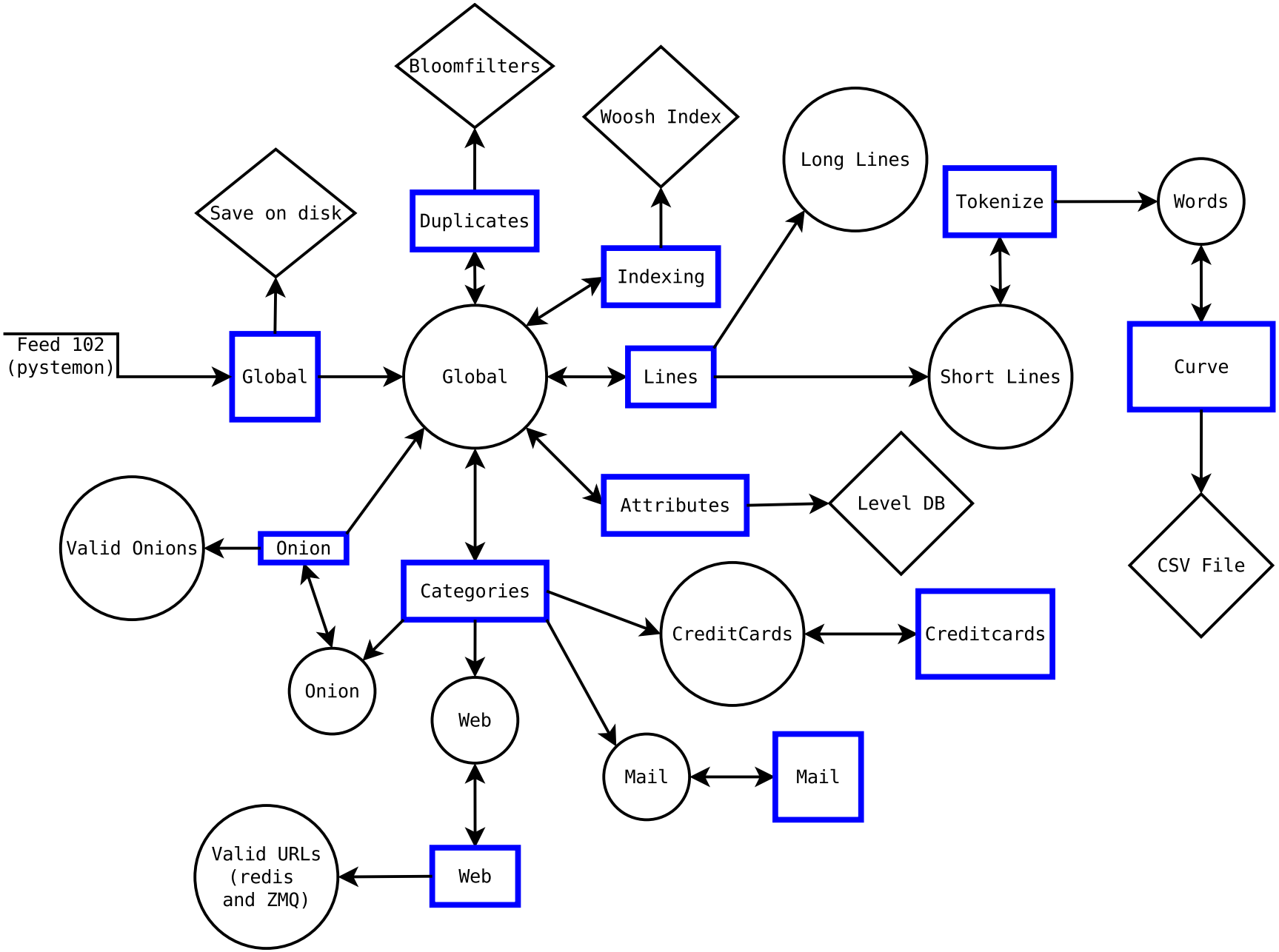
- Pattern matching is good for known information leaks...
- ... but not enough for proactive detection
- New trends

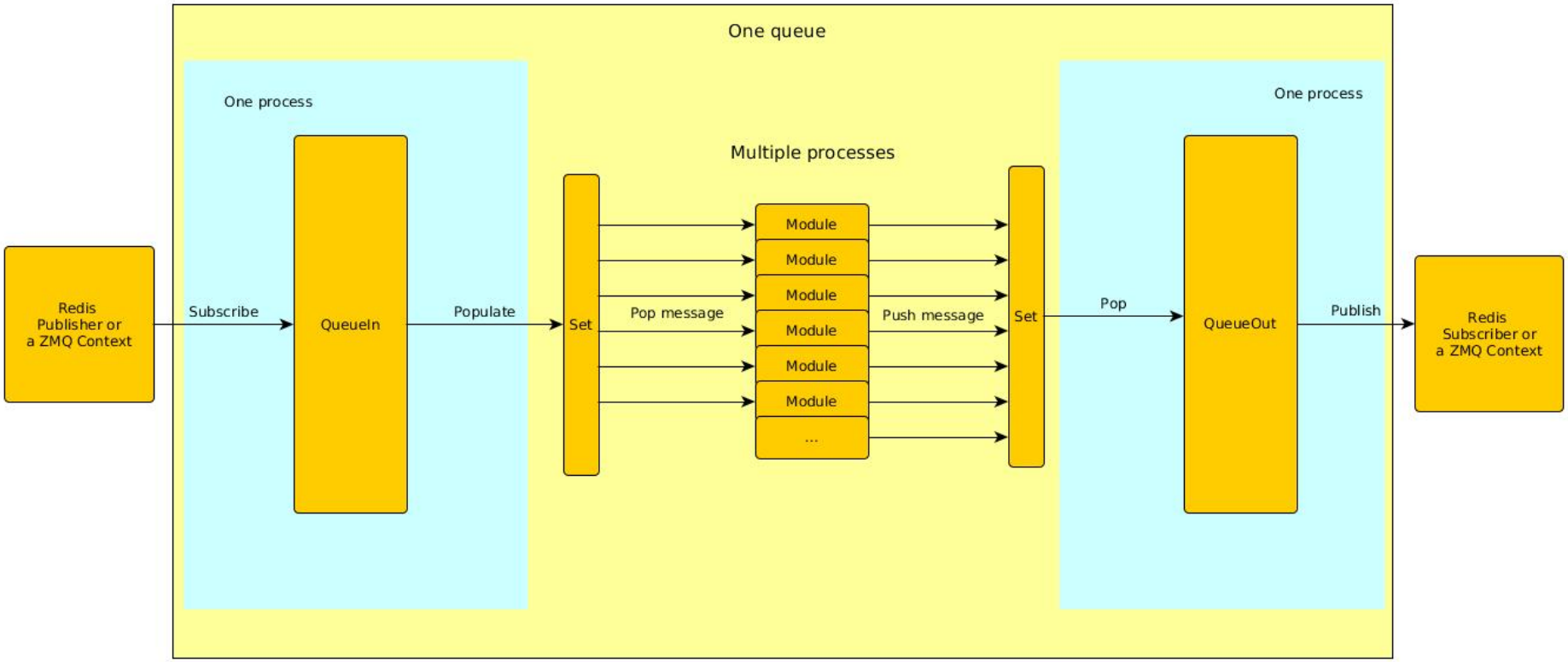
Monitoring - Problems

- Duplicate
- Backlog import
- Size of the database
- Access time
- Real time (1 Paste/second)
- Flexibility
  - New Keywords
  - New Module

Monitoring - Current implementation

- Workflow based on queues
- Many simple modules
- Multiprocessing
- Supports Zero MQ and Redis PubSub


Bloomfilters
Woosh Index
Long Lines
Tokenize
Words
Save on disk
Duplicates
Indexing
Feed 102
(pystemon)
Global
Global
Lines
Short Lines
Curve
Attributes
Level DB
Valid Onions
Onion
Categories
CreditCards
Creditcards
CSV File
Onion
Web
Mail
Mail
Valid URLs
(redis
and ZMQ)
Web


One queue
One process
One process
Multiple processes
Redis
Publisher or
a ZMQ Context
Subscribe
QueueIn
Populate
Set
Pop message
Module
Module
Module
Module
Module
Module
...
Push message
Set
Pop
QueueOut
Publish
Redis
Subscriber or
a ZMQ Context

Monitoring - Framework implementation

1. Add your module in bin/packages/modules.cfg
2. Use bin/template.py as a sample and create file in bin/ with the same name as in the config file

Monitoring - Framework implementation

```
import time
from pubsublogger import publisher
from Helper import Process

def do_something(message):
    return None

if __name__ == '__main__':
    publisher.port = 6380
    publisher.channel = 'Script'
    config_section = '<section name>'
    p = Process(config_section)
    publisher.info("<description of the module>")

    while True:
        message = p.get_from_set()
        if message is None:
            publisher.debug("{} queue is empty, waiting".format(config_section))
            time.sleep(1)
            continue

        something_has_been_done = do_something(message)
        p.populate_set_out(something_has_been_done)
```

Analysis - Existing modules

- Full text indexing
- Attribute (size, mimetype, date...)
- Valid URL, Onion-Website and Email-Address
- Valid Credit card (Luhn-Algorithm)

Analysis - New modules

- Find keys (public and private), OpenPGP, SSL, ...
- Unknown pastebin URLs (private pastes?)
- Base64 encoded
- Emails + attachements?
- Duplicates
- Encrypted blobs in ascii $\rightarrow$ gpg
- Entropy of files / binaries, encrypted?
- Webshells (e.g. evals)
- Language detection, colors
- Chat IRC + politeness/sentiment analysis
- Topic of the paste
- Use john the ripper on hashes

Install & config

- Create a Github account
- Fork `https://github.com/CIRCL/AIL-framework`
- Install the framework (look at .travis.yml) - Run installing_deps.sh
- Make sure you're always in the virtual env - Run . AILENV/bin/activate
- Check the config - bin/packages/config.cfg
- Go to ./bin/
  - Launch the servers: launch_redis.sh, launch_lvldb.sh
  - Launch the logging systems: launch_logs.sh (if it fails, check if log_subscriber is executable
  - Launch the queues: launch_queues.py
  - Launch the scripts - launch_scripts.sh
  - Import the files to classify - import_dir.py
- Look at the logs - logs/*
- Write your own module.

Conclusion

- Source (License AGPL):
  `https://github.com/CIRCL/AIL-framework`
- Contact / Questions / Bugs:
  `https://github.com/CIRCL/AIL-framework/issues`
- E-Mail: info@circl.lu - CA57 2205 C002 4E06 BA70 BE89 EAAD CFFC 22BD 4CD5