Collecting, monitoring and analzing unstructured data

ADaCoR (Advanced Data Collection and Risks) Industry Workshop

CIRCL
Computer Incident
Response Center
Luxembourg

TLP:WHITE

AIL framework - Framework for Analysis of Information Leaks

April 19, 2016

Overview

Collecting leaks: Pastebin and friends
- Motivation
- Evolution
- Challenges
- Current implementation

Handling leaks - open problems

AIL framework implementation

Conclusion

Introduction to Pastebin

From Wikipedia[1]:

- Web application to publish texts
- Time restricted publications
- Available via a developer interface
- Anonymously accessible

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`

Introduction to 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

Motivation

Point of view of the attacker

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

Leak collection - motivation

Point of view of the analyst

- Information source of
  - Databases
  - Credit cards
  - Login information (passwords, keys, ...) $\rightarrow$ compromised systems
- Very often data concerning organisations in our constituency

Collecting leaks

Evolution

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 ($\sim$ 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`

Collecting leaks

Challenges

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

Collecting leaks

Current implementation

- New proxy list every day
- API queries through the proxys
- Error messages analysis (Socket, Timeout, Proxy, Temporary Ban, Blacklist)
- Reliability tests (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

Collecting leaks

Current implementation

```
[I] Proxy status: 8 proxies left in memory
[F] Proxy 182.118.23.7:8081 fail count: 3/3
[F] Removing proxy 182.118.23.7:8081 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
```

Handling leaks - Why is Pystemon not enough?

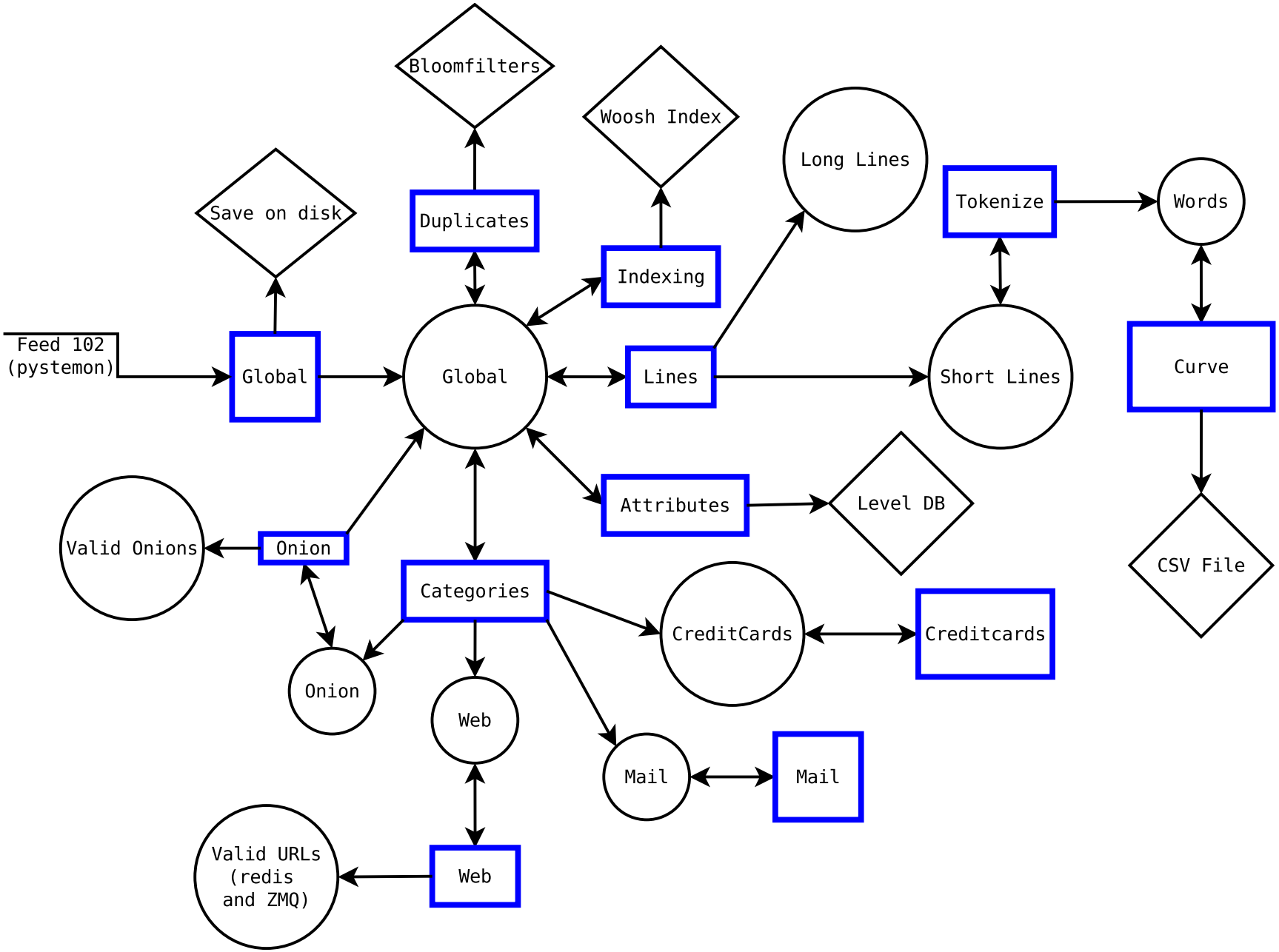
- Pattern matching is good for known information leaks...
- ... but not enough for proactive detection
- New trends

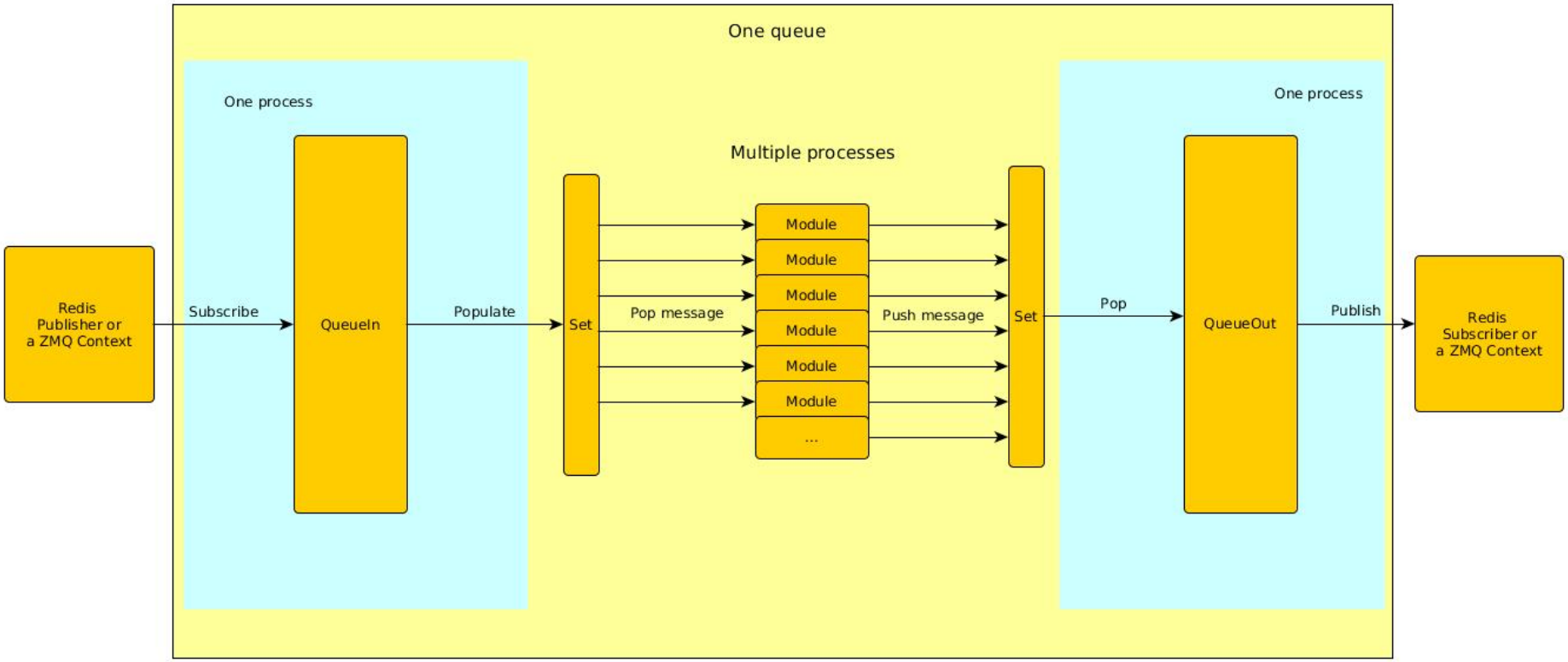
Handling leaks - problems

- Duplicates $\rightarrow$ repostings
- Unknown / mixed data structures
- Real time (1 Paste/second)
- Flexibility
  - New keywords
  - New data type $\rightarrow$ new module

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
CSV File
CreditCards
Creditcards
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

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

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)
```

Existing analysis modules

- Full text indexing
- Attribute (size, mimetype, date...)
- Valid URL, Onion-Website and Email-Address
- Valid Credit card (Luhn-Algorithm)
- Encrypted blobs in ascii $\Rightarrow$ gpg
- Phone number detection
- Source code identification
- Ransomware text detection
- Credentials identification

Future analysis modules

- Find keys (public and private), GPG, SSL, ...
- Unknown pastebin URLs (private pastes?)
- Base64 encoded
- Emails + attachements?
- Duplicates
- Webshells (evals)
- Language
- Chat IRC + politeness
- Topic of the paste

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