IOCs are Dead, Long live the IOCs!

Getting Started IntelMQ

HACK.LU
2019

Who I am?

Celine Massompierre

Incident Handler – Excellium-Services CSIRT

- Almost 10 years as Business Intelligence Analyst
- Newbie in security field (~ 3 years)
- Enjoy learning and sharing new things

⚠ Not an expert, not a core developer of IntelMQ. Just a user :)

Newbie Course

A word about Threat Intel

Threat Intel

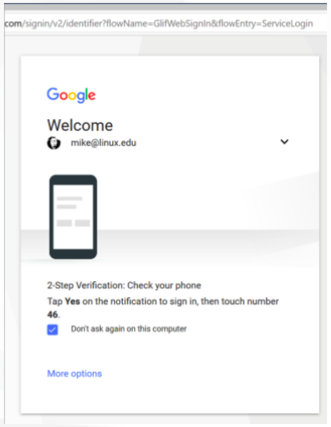
Threat Intelligence is one of these trendy words in the security world...

But what it is exactly?

Threat Intelligence is a way of detecting and avoiding bad things.

Threat?

Threats could be malware, botnet, ransomware, exploit, theft….



com/signin/v2/identifier?flowName=GlifWebSignIn&flowEntry=ServiceLogin

Google

Welcome

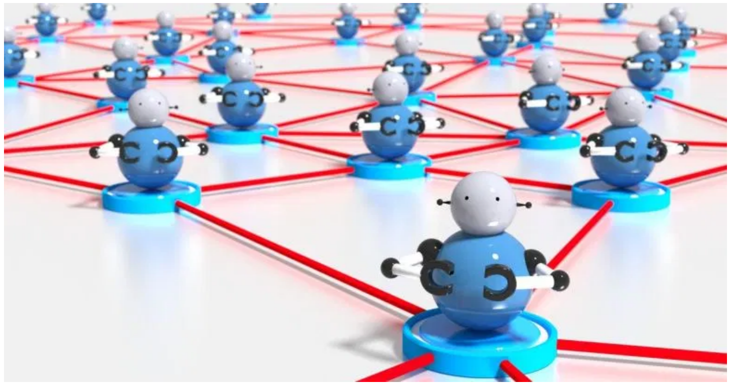
mike@linux.edu

2-Step Verification: Check your phone

Tap **Yes** on the notification to sign in, then touch number **46**.

☑ Don't ask again on this computer

More options



How?

Using Indicator Of Compromise (IOC)

IOCs are artefacts which identify something in a clear and an unambiguous way.

For example:

Some malware contains url or ip address hard-coded for reaching their command & control. These artifacts are IOCs.

IOC Feeds

These lists are available through to many providers.

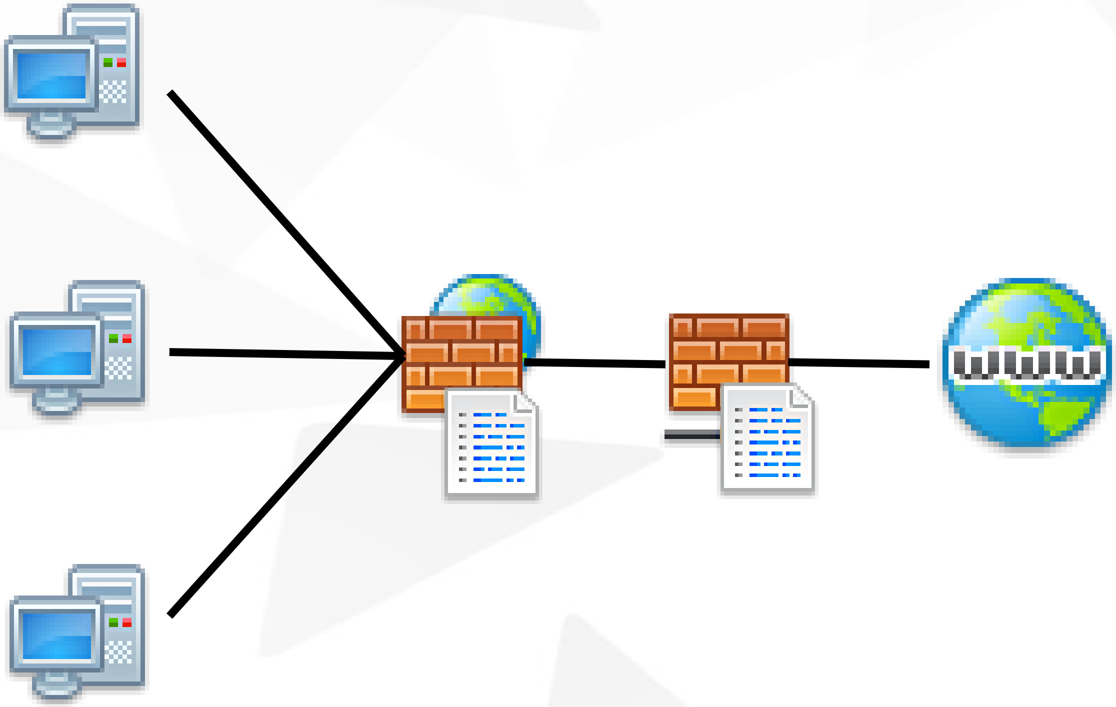
Most of the time they are related to Network artifact.

For example:

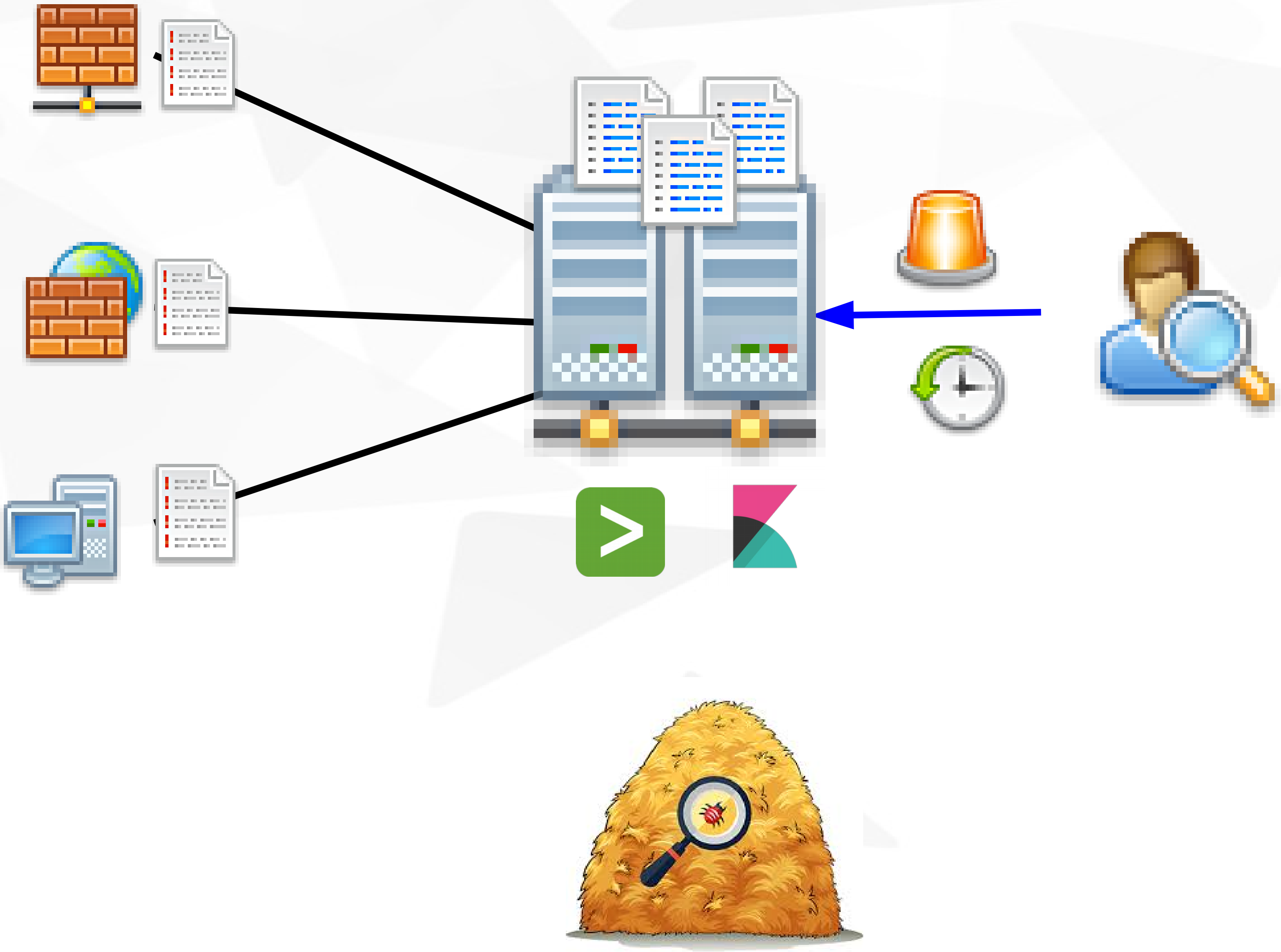
IP addresses used by Feodo network are listed by abuse.ch:
https://feodotracker.abuse.ch

IP addresses attacking specific service (ftp, imap, mail…) are listed by blocklist.de:
https://www.blocklist.de/en/export.html

Then What?



Then What?



IntelMQ
How does this work?

HACK.LU
2019

IntelMQ

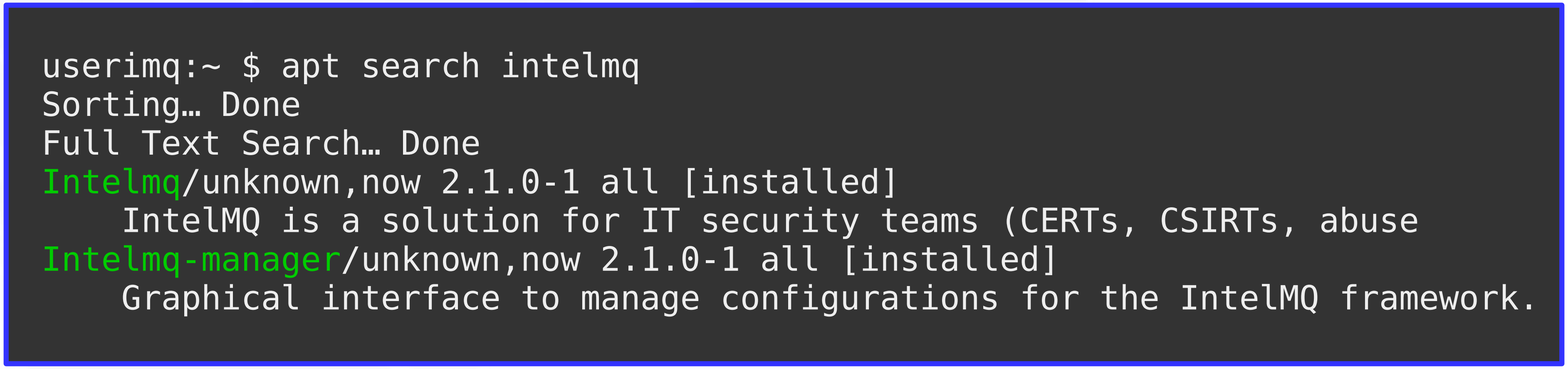
https://github.com/certtools/intelmq

- Tool for gathering, cleaning and enriching IOCs
- Deals with many different sources and destinations.
- Easy to install (package), manage and improve
- Open Source Project
- Created by multiple CERTs (Trusted Introducer*) and maintained by CERT.AT

IntelMQ = Threat **Intel** feeds + **M**essage **Q**ueueing system

* https://www.trusted-introducer.org/

IntelMQ

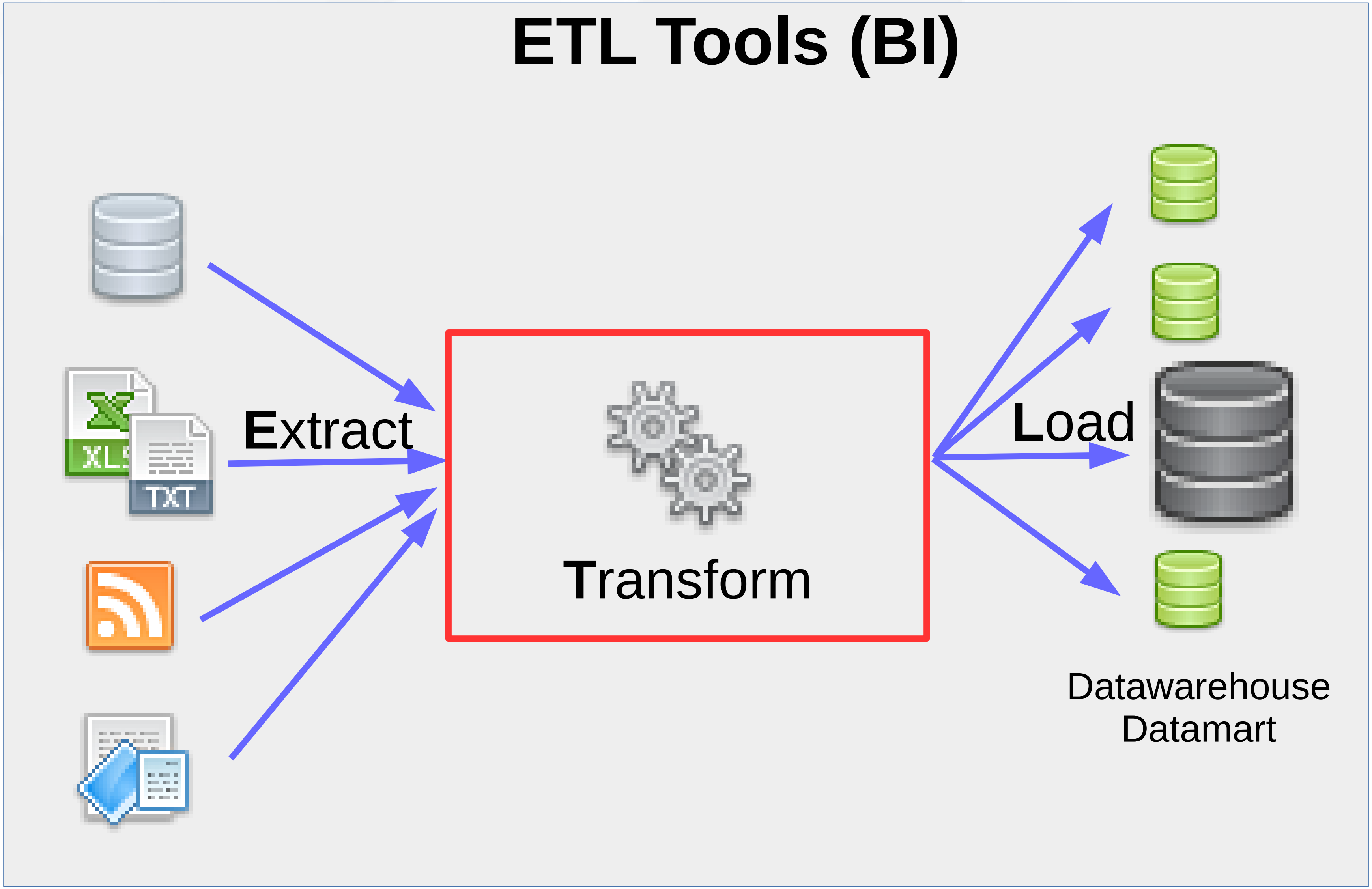


```
userimq:~ $ apt search intelmq
Sorting… Done
Full Text Search… Done
Intelmq/unknown,now 2.1.0-1 all [installed]
    IntelMQ is a solution for IT security teams (CERTs, CSIRTs, abuse
Intelmq-manager/unknown,now 2.1.0-1 all [installed]
    Graphical interface to manage configurations for the IntelMQ framework.
```

- IntelMQ is a command line tool.
- IntelMQ-Manager is a must do for development.

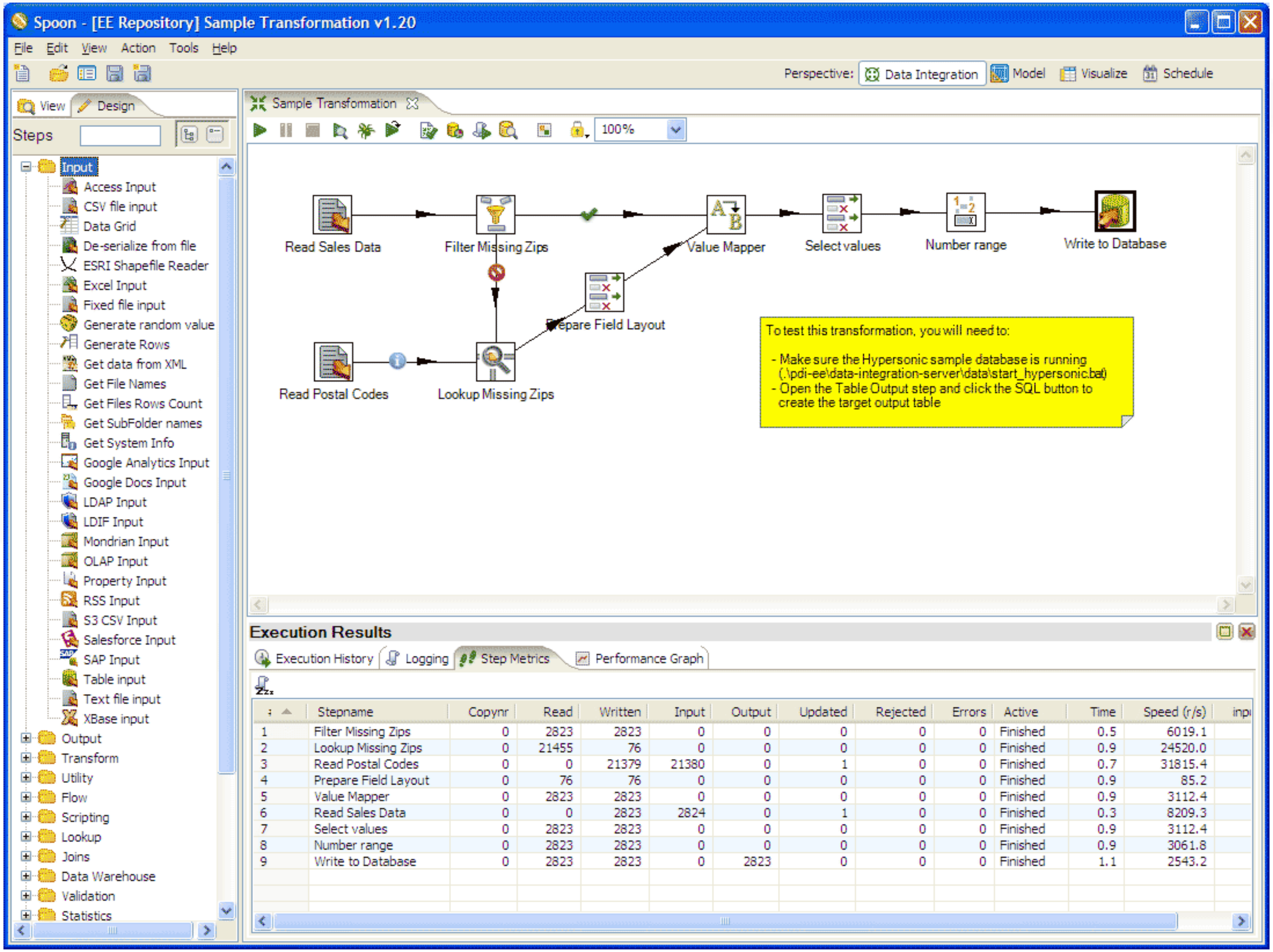
⚠ Avoid IntelMQ-Manager in production without enhance security!

Spirit


ETL Tools (BI)
XLS
TXT
Extract
Transform
Load
Datawarehouse
Datamart

Spirit

One example of ETL: Pentaho kettle

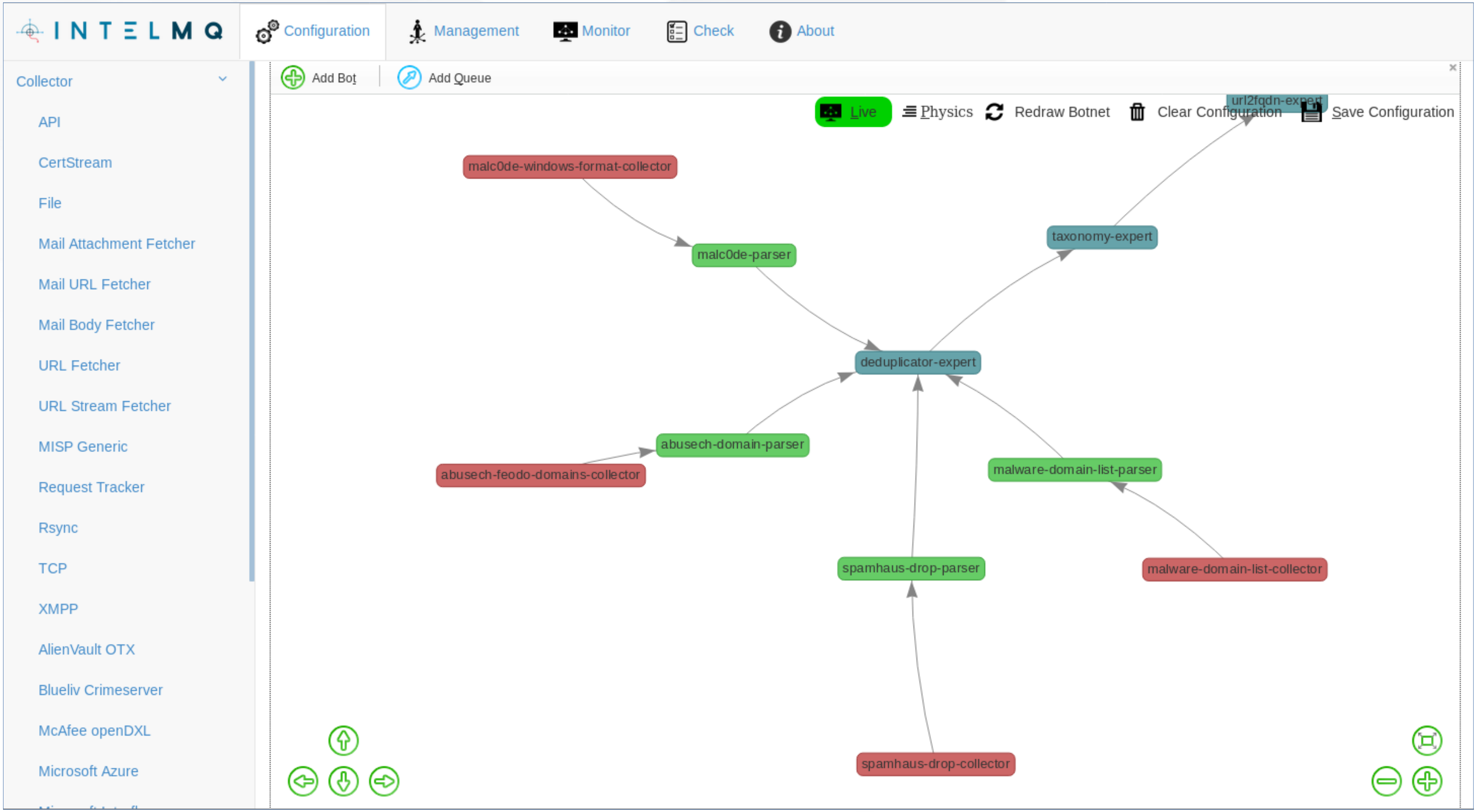


To test this transformation, you will need to:

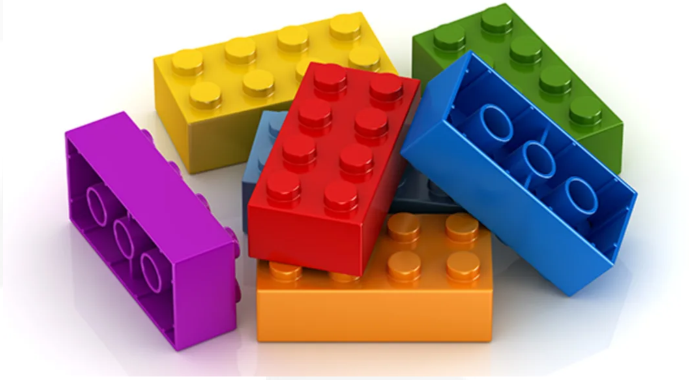
- Make sure the Hypersonic sample database is running (.\pdi-ee\data-integration-server\data\start_hypersonic.bat)
- Open the Table Output step and click the SQL button to create the target output table

#	Stepname	Copynr	Read	Written	Input	Output	Updated	Rejected	Errors	Active	Time	Speed (r/s)	inpu
1	Filter Missing Zips	0	2823	2823	0	0	0	0	0	Finished	0.5	6019.1	
2	Lookup Missing Zips	0	21455	76	0	0	0	0	0	Finished	0.9	24520.0	
3	Read Postal Codes	0	0	21379	21380	0	1	0	0	Finished	0.7	31815.4	
4	Prepare Field Layout	0	76	76	0	0	0	0	0	Finished	0.9	85.2	
5	Value Mapper	0	2823	2823	0	0	0	0	0	Finished	0.9	3112.4	
6	Read Sales Data	0	0	2823	2824	0	1	0	0	Finished	0.3	8209.3	
7	Select values	0	2823	2823	0	0	0	0	0	Finished	0.9	3112.4	
8	Number range	0	2823	2823	0	0	0	0	0	Finished	0.9	3061.8	
9	Write to Database	0	2823	2823	0	2823	0	0	0	Finished	1.1	2543.2	

InteMQ-Manager



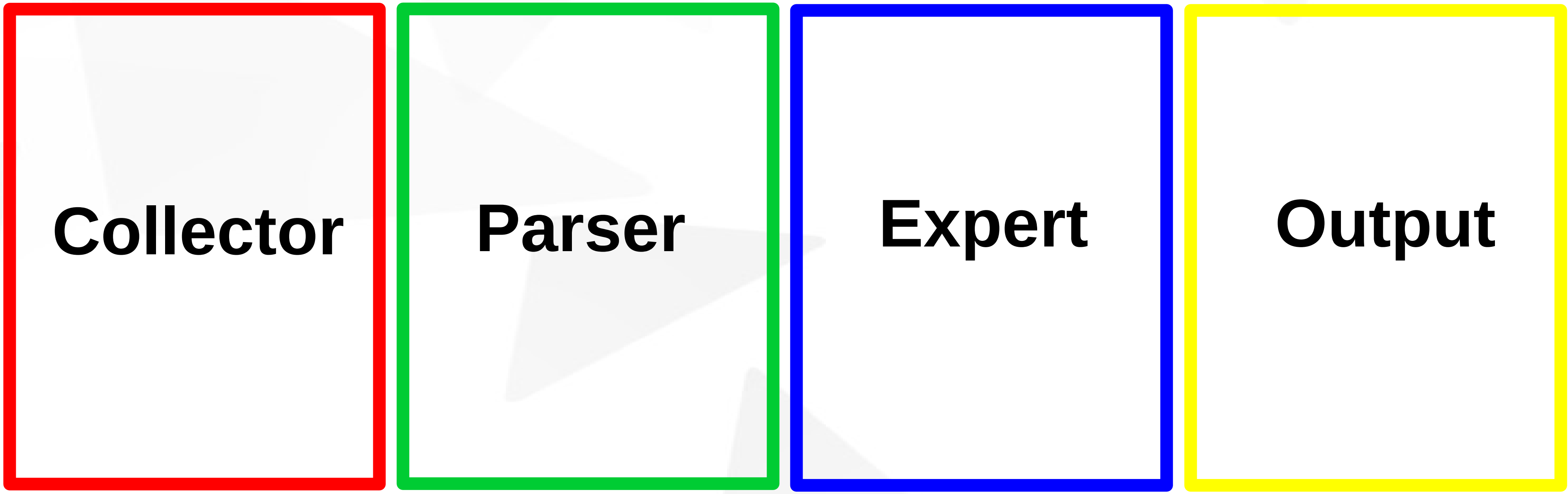
Important Concept



Bots

In IntelMQ, you manipulate bots and arrange them to create your own data flow.
A bot is a kind of “object” which does one thing.

Bots are divided in 4 groups:



Collector

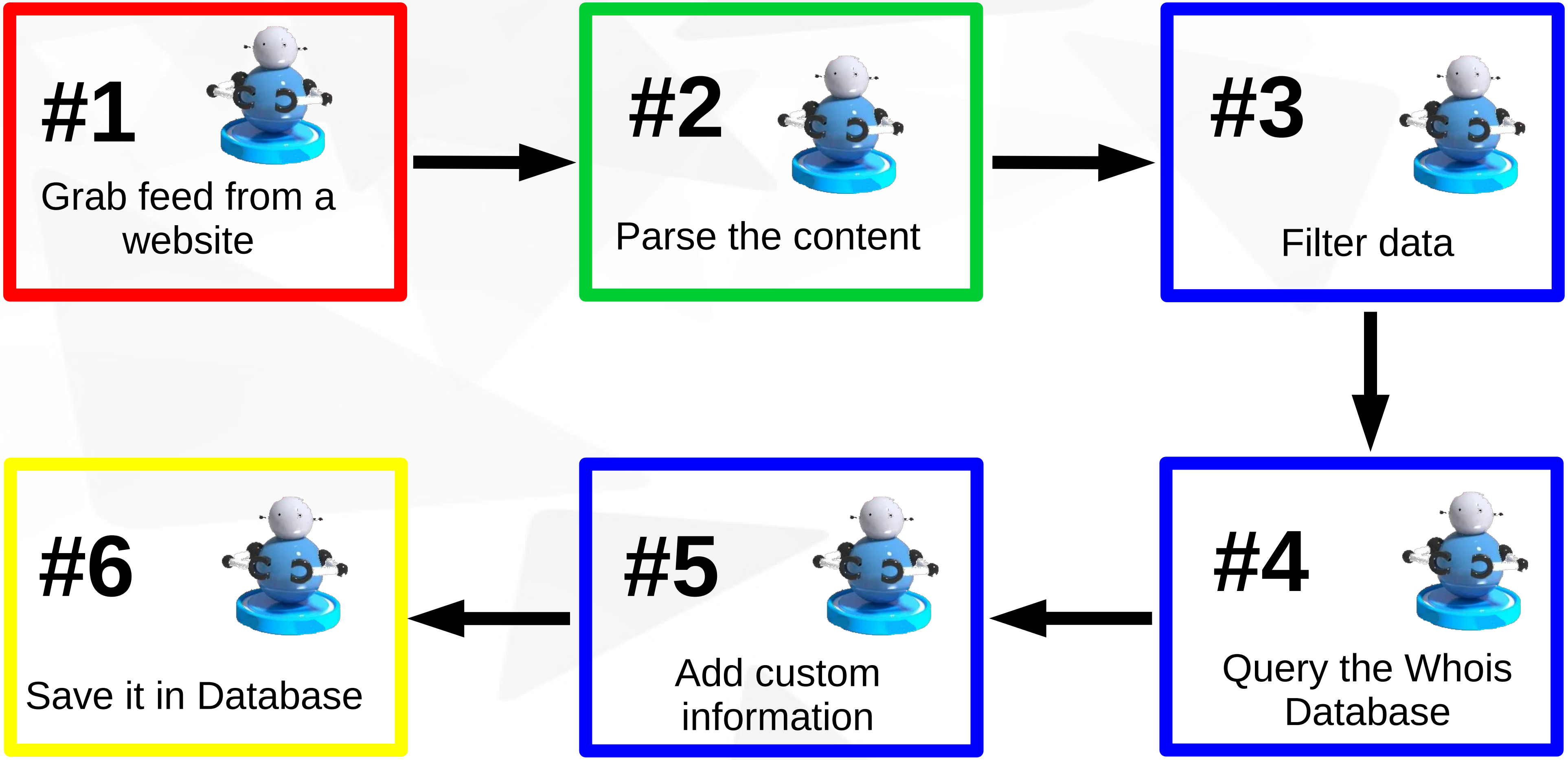
Parser

Expert

Output

Bots

Example of Data flow



#1 Grab feed from a website

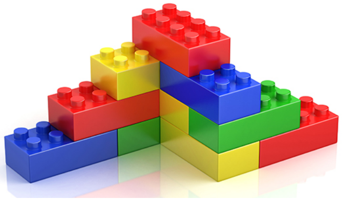
#2 Parse the content

#3 Filter data

#4 Query the Whois Database

#5 Add custom information

#6 Save it in Database



Let’s play!

VM

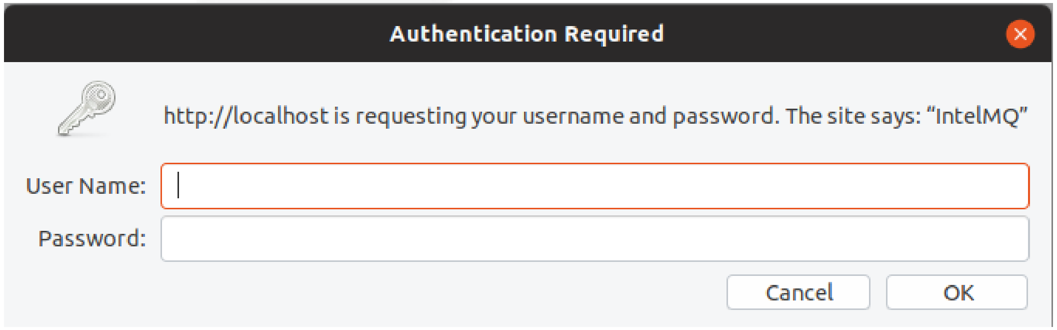
Ubuntu: user / User123

US KEYBOARD

US

In command line, use ‘`setxkbmap`’ for changing it.
For example, for french: ‘`setxkbmap fr`’

IntelMQ-Manager (Firefox): admin / Admin123



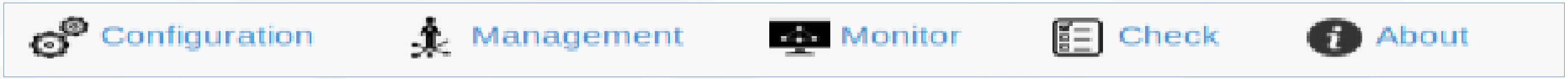
Authentication Required

http://localhost is requesting your username and password. The site says: “IntelMQ”

User Name:

Password:

IntelMQ-Manager



Design of the data flow

Bots Management
Start/stop

Debug and log access

Status of installation

Self explained, right?

Exercises

All exercises are available in the folder IOCs on the Desktop.

- exo.txt: statement of each exercise with url of the feeds
- deploy.sh: deploy configuration files from ‘actual’ folder to intelmq folder
- start.sh and stop.sh: start and stop intelMQ and IntelMQ-Manager

- empty folder: empty configuration files, if you want to restart
- default folder: default example provided with intelmq
- file-output folder: default location for the output file
- solutions folder: solution of each exercises if you get lost

Exercise 00

Get feodo tracker blocklist

Guided

In this exercise, we create a simple data flow with 3 bots: 1 collector, 1 parser and 1 output

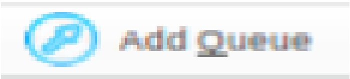
1. Drag and drop the following bots:

- Collector > Mail URL Fetcher
  http_url: https://feodotracker.abuse.ch/downloads/ipblocklist.txt

- Parser > Abuse.ch IP

- Output > File
  file: let the default value

2. Use add Queue button for linking these 3 bots



Click, then hold the click on the first bots and release it on the next

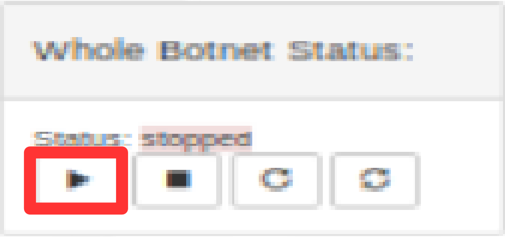
3. Save configuration

Exercise 00

Get feodo tracker blocklist

Guided

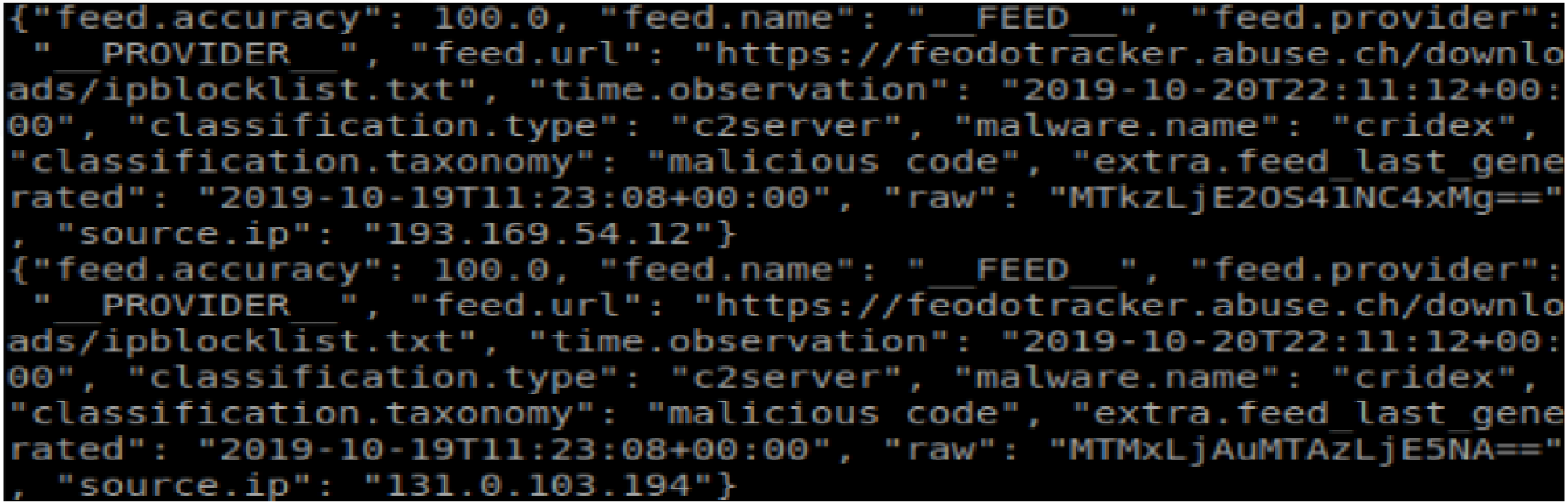
On Management tab, start all the bots:



Whole Botnet Status:

Status: stopped

Take a look at the target file > output-file/event.txt



```
{"feed.accuracy": 100.0, "feed.name": "__FEED__", "feed.provider":
 "__PROVIDER__", "feed.url": "https://feodotracker.abuse.ch/downlo
ads/ipblocklist.txt", "time.observation": "2019-10-20T22:11:12+00:
00", "classification.type": "c2server", "malware.name": "cridex",
"classification.taxonomy": "malicious code", "extra.feed_last_gene
rated": "2019-10-19T11:23:08+00:00", "raw": "MTkzLjE2OS41NC4xMg=="
, "source.ip": "193.169.54.12"}
{"feed.accuracy": 100.0, "feed.name": "__FEED__", "feed.provider":
 "__PROVIDER__", "feed.url": "https://feodotracker.abuse.ch/downlo
ads/ipblocklist.txt", "time.observation": "2019-10-20T22:11:12+00:
00", "classification.type": "c2server", "malware.name": "cridex",
"classification.taxonomy": "malicious code", "extra.feed_last_gene
rated": "2019-10-19T11:23:08+00:00", "raw": "MTMxLjAuMTAzLjE5NA=="
, "source.ip": "131.0.103.194"}
```

Exercise 00

Guided

Get feodo tracker blocklist

IN

```
# DstIP
78.46.103.90
94.177.216.217
69.163.33.84
131.0.103.200
120.138.101.250
186.71.150.23
31.128.13.45
192.3.104.40
51.89.115.120
66.85.156.81
144.91.76.214
194.36.189.165
194.5.250.98
...
```

OUT

```
{
        “feed.accuracy”: 100,
        “feed.name”: “__FEED__”,
        “feed.provider”: “__PROVIDER__”,
        “feed.url”: “https://feodotracker.abuse.ch/downloads/ipblocklist.txt”,
        “time.observation”: “2019-10-20T22:11:12+00:00”,
        “classification.type”: “c2server”,
        “malware.name”: “cridex”
        “classification.taxonomy”: “malicious code”,
        “extra.feed_last_generated”: “2019-10-19T11:23:08+00:00”,
        “raw”: “NzguNDYuMTAzLjkw”,
        “source.ip”: “78.46.103.90”
}
```

Normalization

The main goal of IntelMQ is to automate gathering, and also to normalize and enrich IOCs.

IntelMQ provide a predefined list of target fields:

https://github.com/certtools/intelmq/blob/develop/docs/Harmonization-fields.md

These fields are divided in multiple “group”: feed, source, destination, time, classification...

And speaking of classification, IntelMQ use an extended version of eCSIRT II taxonomy:

https://www.trusted-introducer.org/Incident-Classification-Taxonomy.pdf

https://github.com/certtools/intelmq/blob/develop/docs/Data-Harmonization.md

Under the hood, fields are defined in the harmonization.conf file

Exercise 01

Get blocklists as it

1. Based on the previous exercise, change the feed.provider and feed.name.

2. Try to add another simple data flow (1 collector, 1 parser and 1 output)
   Feel free to test anything, as long as you find a feed :)

   For helping you, look at this page:
   https://github.com/certtools/intelmq/blob/master/docs/Feeds.md

3. Some feeds that you could try:
   https://feodotracker.abuse.ch/downloads/ipblocklist.csv
   https://ransomwaretracker.abuse.ch/feeds/csv
   https://www.openphish.com/feed.txt

Exercise 02

Let’s play with generic CSV parser

If a csv feed does not have a dedicated parser, you could use the generic csv parser.

For helping you, look at this page:
https://github.com/certtools/intelmq/blob/develop/docs/Bots.md#generic-csv-parser

1. Try to parse feodo tracker (ipblocklist.csv) with the generic csv parser

Tip #1: You need to use harmonization.conf for naming destination field.
Available in IOCs/default folder

Tip #2: Some source field are multi purpose
Identify different field and separate them with a |
Ex: source.url|source.fqdn|source.ip

Tip #3: When a source field can be empty, use __IGNORE__ keyword
Ex: source.url|__IGNORE__

Exercise 02

Let’s play with generic CSV parser

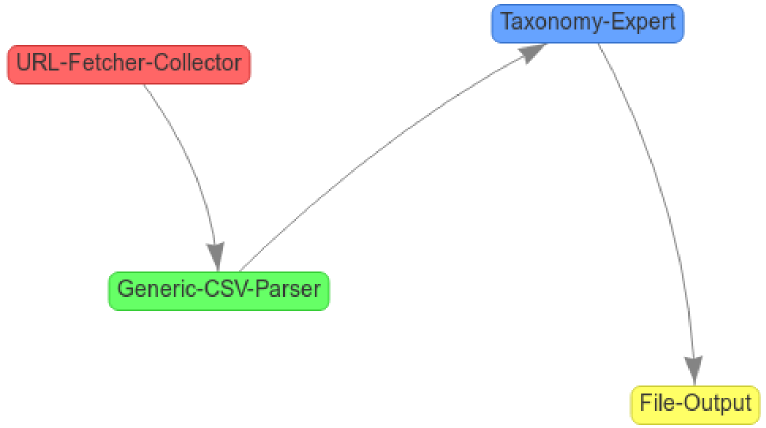
```
{
    "classification.type": "c2server",
    "destination.ip": "186.47.122.182",
    "destination.port": 449,
    "feed.accuracy": 100.0,
    "feed.name": "__FEED__",
    "feed.provider": "__PROVIDER__",
    "feed.url": "http://localhost/downloads/ipblocklist.csv",
    "malware.name": "trickbot",
    "raw":
"MjAxOS0xMC0yMSAxMDowNTowNCwyMDAuMTI3LjEyMS45OSw0NDksLFRyaWNrQm90DQ0K
MjAxOS0xMC0yMSAxMDowNTowNCwxODYuNDcuMTIyLjE4Miw0NDksLFRyaWNrQm90DQo=",
    "time.observation": "2019-10-22T09:22:22+00:00",
    "time.source": "2019-10-21T10:05:04+00:00"
}
```

“classification.taxonomy” missing!

Exercise 02

Let’s play with generic CSV parser

For adding “classification.taxonomy”, simply use the “Taxonomy” bot (Expert)

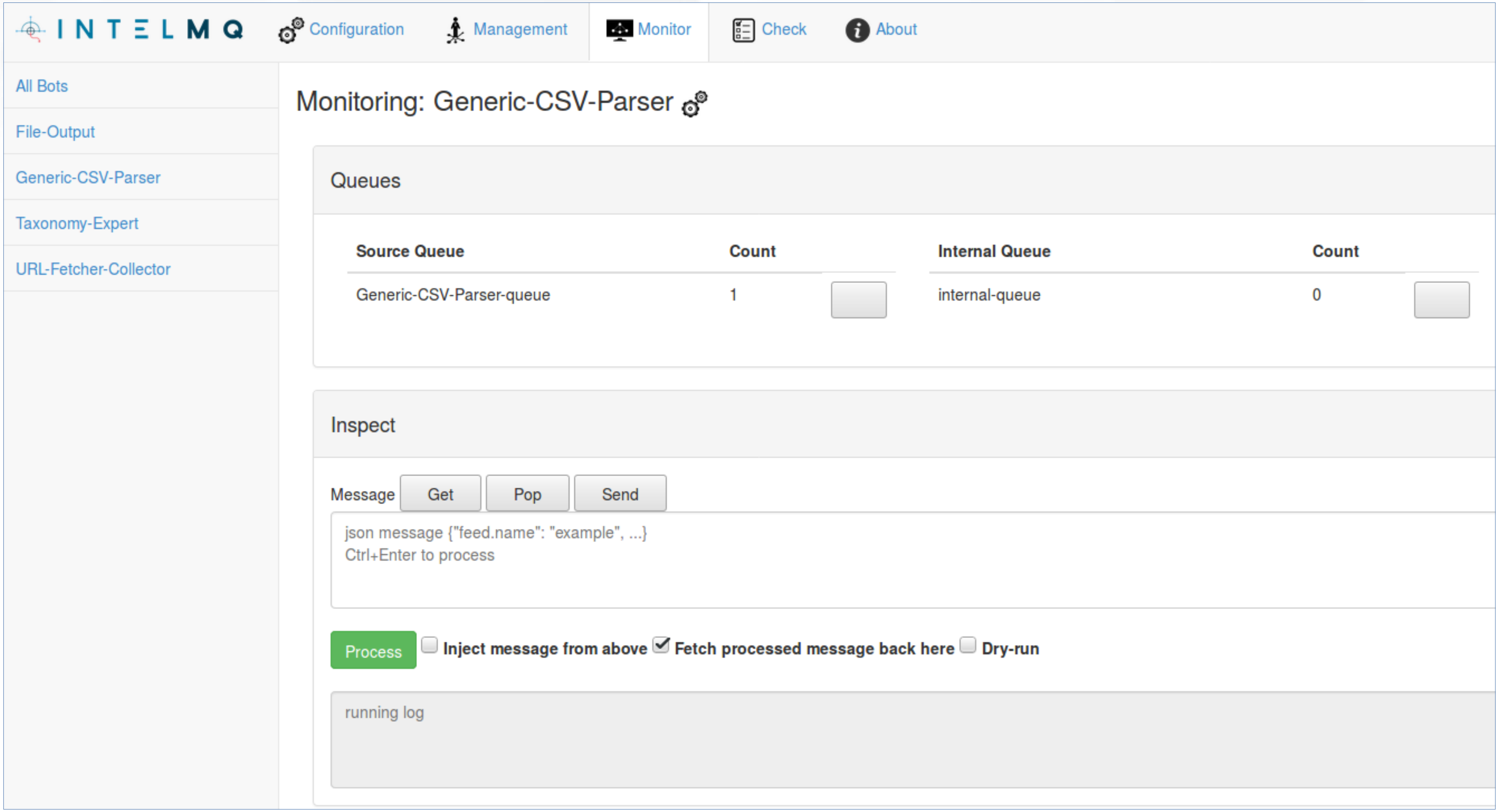


If you don’t have the classification.taxonomy, use script stop.sh and start.sh

Debug

When you want to debug one of the data flow, you can use intelmq-manager.

Stop all bots, then start only the collector, and look at the Parser:



Monitoring: Generic-CSV-Parser

Queues

Source Queue	Count	Internal Queue	Count
Generic-CSV-Parser-queue	1	internal-queue	0

Inspect

Message

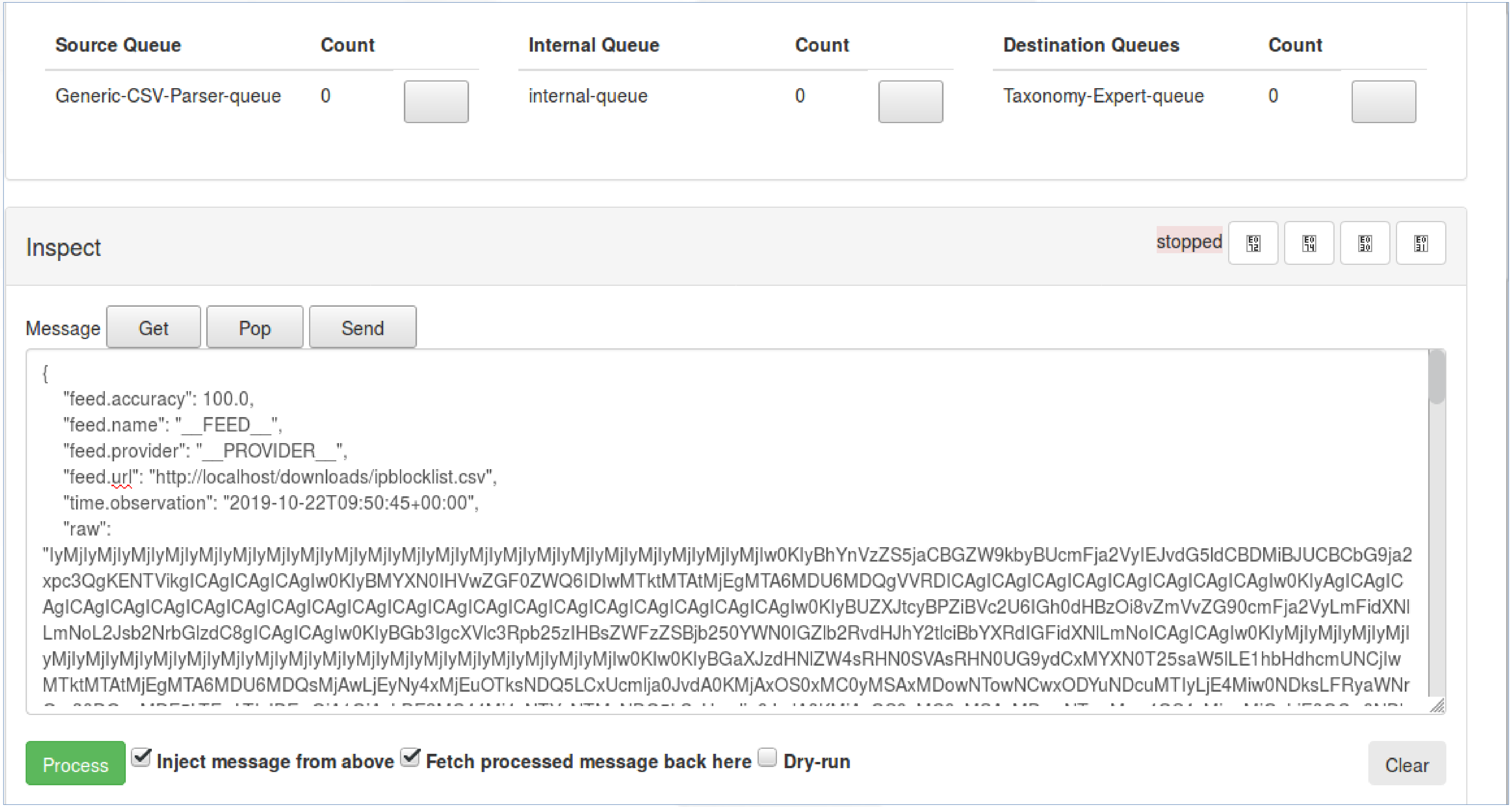
json message {"feed.name": "example", ...}
Ctrl+Enter to process

□ Inject message from above ☑ Fetch processed message back here □ Dry-run

running log

Debug

Pop record in the pipe (pop button) and you get what the collector sent.



Source Queue	Count		Internal Queue	Count		Destination Queues	Count	
Generic-CSV-Parser-queue	0		internal-queue	0		Taxonomy-Expert-queue	0	

Inspect

stopped

Message

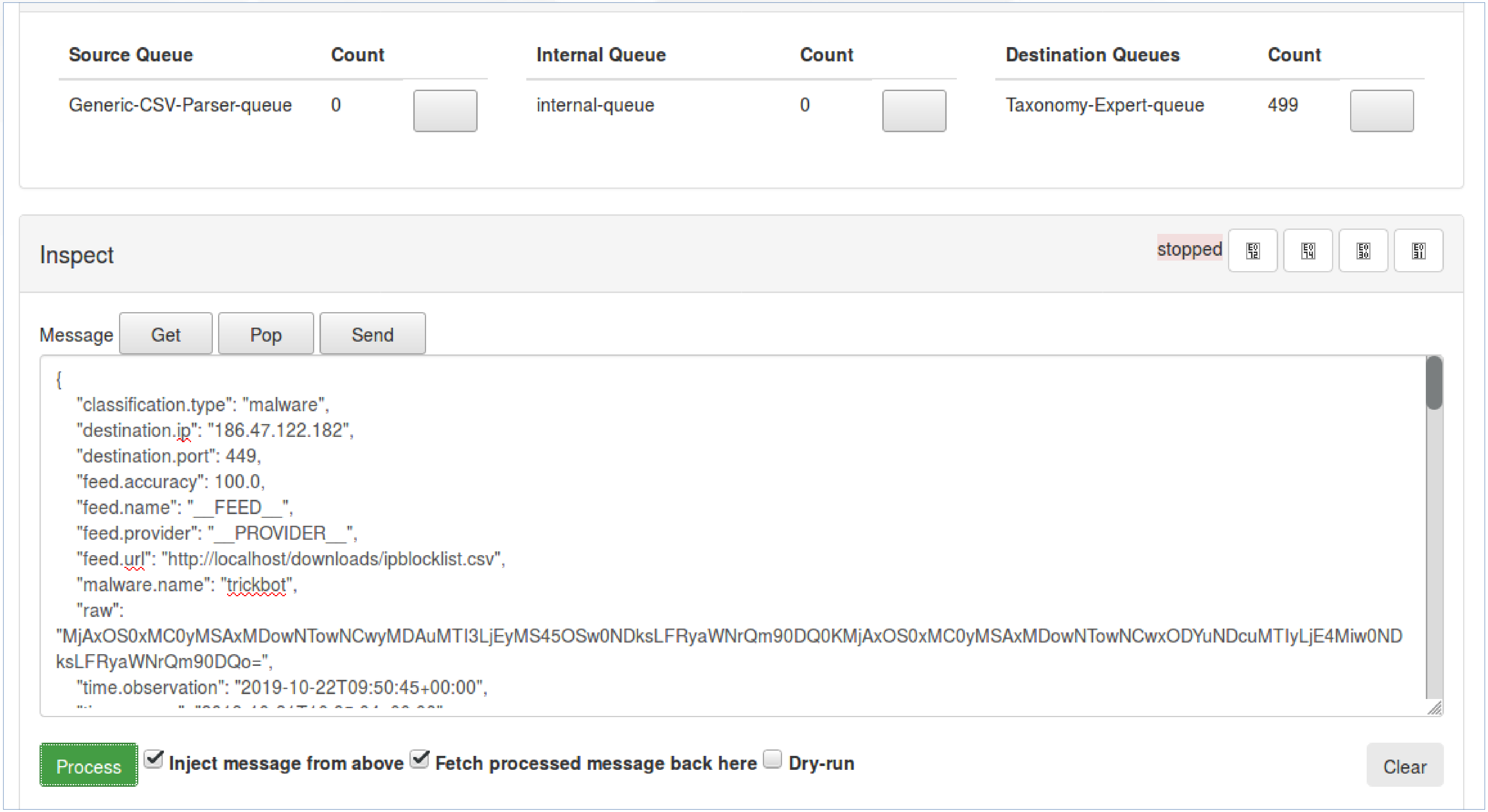
```
{
    "feed.accuracy": 100.0,
    "feed.name": "__FEED__",
    "feed.provider": "__PROVIDER__",
    "feed.url": "http://localhost/downloads/ipblocklist.csv",
    "time.observation": "2019-10-22T09:50:45+00:00",
    "raw":
"IyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIw0KIyBhYnVzZS5jaCBGZW9kbyBUcmFja2VyIEJvdG5ldCBDMiBJUCBCbG9ja2
xpc3QgKENTVikgICAgICAgICAgIw0KIyBMYXN0IHVwZGF0ZWQ6IDIwMTktMTAtMjEgMTA6MDU6MDQgVVRDICAgICAgICAgICAgICAgICAgICAgICAgIw0KIyAgICAgIC
AgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIw0KIyBUZXJtcyBPZiBVc2U6IGh0dHBzOi8vZmVvZG90cmFja2VyLmFidXNl
LmNoL2Jsb2NrbGlzdC8gICAgICAgIw0KIyBGb3IgcXVlc3Rpb25zIHBsZWFzZSBjb250YWN0IGZlb2RvdHJhY2tlciBbYXRdIGFidXNlLmNoICAgICAgIw0KIyMjIyMjIyMjIyMjI
yMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIw0KIw0KIyBGaXJzdHNlZW4sRHN0SVAsRHN0UG9ydCxMYXN0T25saW5lLE1hbHdhcmUNCjIw
MTktMTAtMjEgMTA6MDU6MDQsMjAwLjEyNy4xMjEuOTksNDQ5LCxUcmlja0JvdA0KMjAxOS0xMC0yMSAxMDowNTowNCwxODYuNDcuMTIyLjE4Miw0NDksLFRyaWNr
```

☑ Inject message from above ☑ Fetch processed message back here □ Dry-run

Base64 is the whole input file content!

Debug

For testing the bot, just use the process button



Source Queue	Count	Internal Queue	Count	Destination Queues	Count
Generic-CSV-Parser-queue	0	internal-queue	0	Taxonomy-Expert-queue	499

Inspect

stopped

Message

```
{
    "classification.type": "malware",
    "destination.ip": "186.47.122.182",
    "destination.port": 449,
    "feed.accuracy": 100.0,
    "feed.name": "__FEED__",
    "feed.provider": "__PROVIDER__",
    "feed.url": "http://localhost/downloads/ipblocklist.csv",
    "malware.name": "trickbot",
    "raw":
"MjAxOS0xMC0yMSAxMDowNTowNCwyMDAuMTI3LjEyMS45OSw0NDksLFRyaWNrQm90DQ0KMjAxOS0xMC0yMSAxMDowNTowNCwxODYuNDcuMTIyLjE4Miw0NDksLFRyaWNrQm90DQo=",
    "time.observation": "2019-10-22T09:50:45+00:00",
```

☑ **Inject message from above** ☑ **Fetch processed message back here** □ **Dry-run**

Exercise 03

Expert

Expert bot helps you to enrich data and clean them.
In these exercise, use the debug.

1. Use the following samples and test them manually on ‘RFC-1918’ bot
   - BadIP.txt
   - BadURL.txt

2. Use the samples in url2fqdn.txt file and test them on ‘url2fqdn’ bot

3. Test the ‘Cymru-Whois-Expert’ bot

4. Test others bot if you have time ;)

Exercise 04

Let’s discover the swiss army knife bot: modify

Often, you need to personalized a little bit what you get:

- Remove a field,
- Switch direction or IOC (because all parser did what you want)
- Split an IOC (url for example)
- Add new field

For that, you can used the modify expert:

https://github.com/certtools/intelmq/blob/master/docs/Bots.md#modify

Exercise 04

Let’s discover the swiss knife bot: modify

This bot is not configurable via intelmq-manager. It is time to use terminal (Finally)!

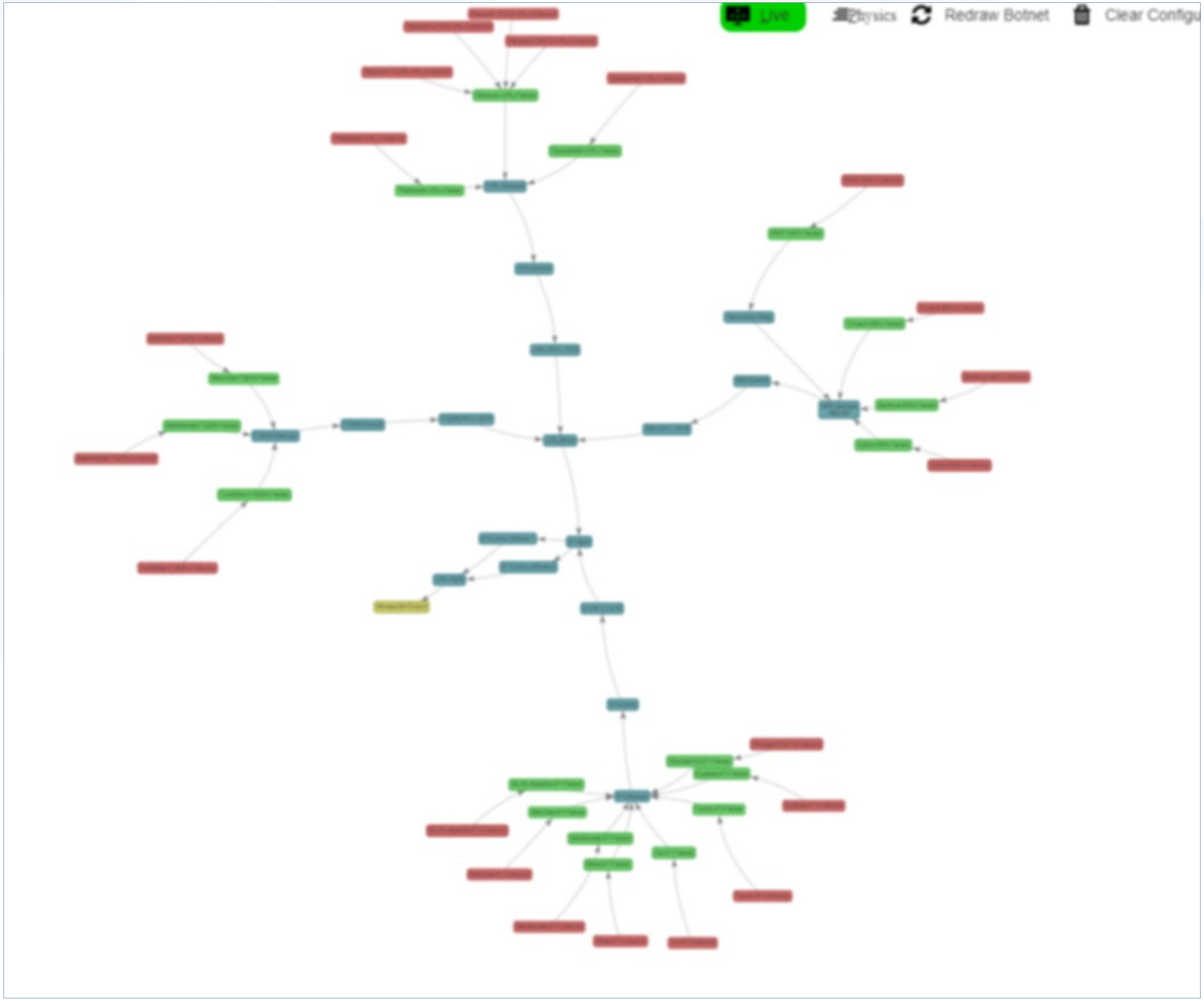
1. Add a new static field

2. Use the modify bot for removing the field accuracy

3. Switch direction of the IOC, from source to destination

4. Use a regex for drop the parameter part of url

GUI is good for...

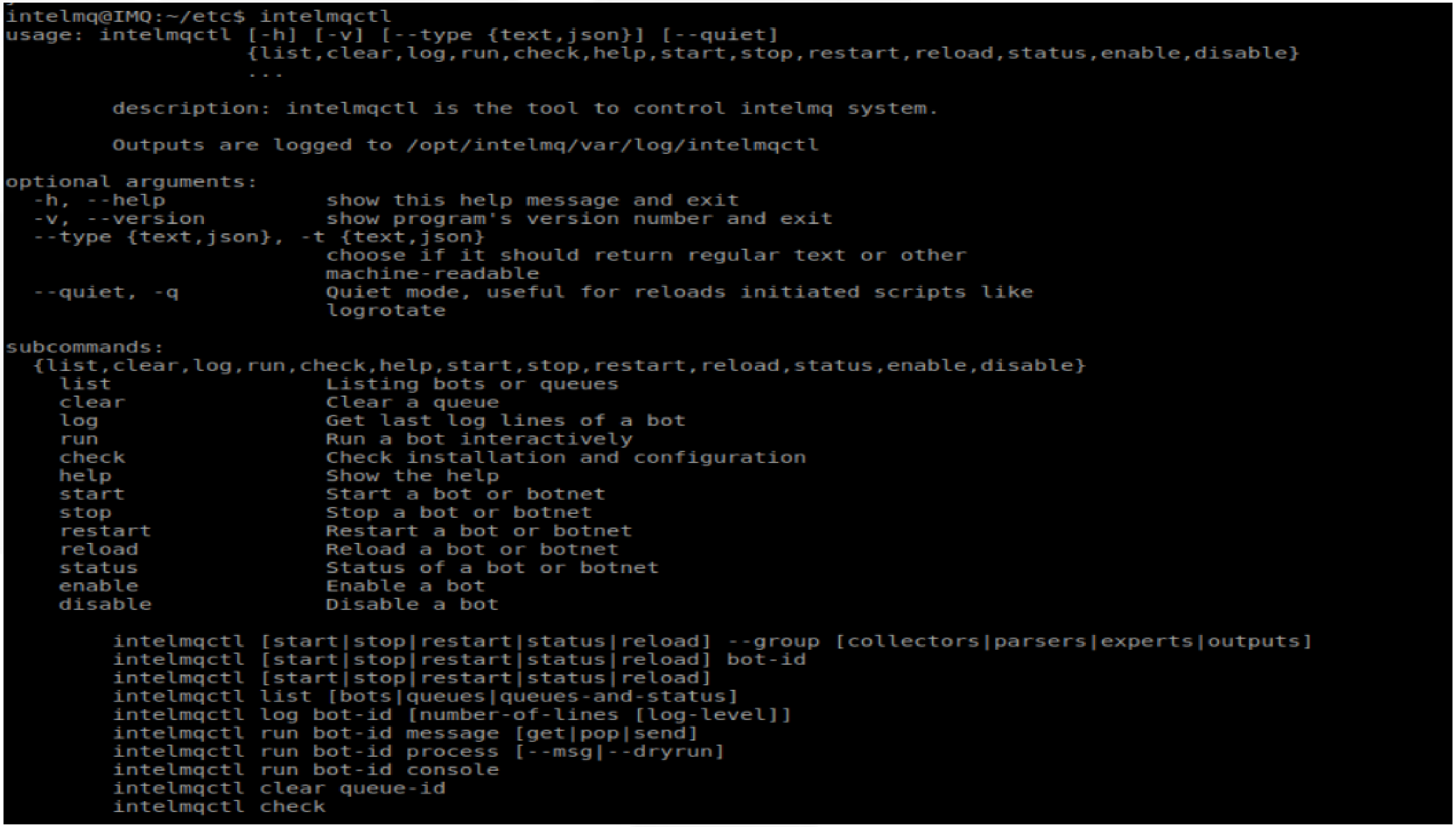


Maintaining this kind of workflow!

Hackers use terminal!

IntelMQ offer a really cool command called intelmqctl.

https://github.com/certtools/intelmq/blob/develop/docs/intelmqctl.md



```
intelmq@IMQ:~/etc$ intelmqctl
usage: intelmqctl [-h] [-v] [--type {text,json}] [--quiet]
                  {list,clear,log,run,check,help,start,stop,restart,reload,status,enable,disable}
                  ...

        description: intelmqctl is the tool to control intelmq system.

        Outputs are logged to /opt/intelmq/var/log/intelmqctl

optional arguments:
  -h, --help            show this help message and exit
  -v, --version         show program's version number and exit
  --type {text,json}, -t {text,json}
                        choose if it should return regular text or other
                        machine-readable
  --quiet, -q           Quiet mode, useful for reloads initiated scripts like
                        logrotate

subcommands:
  {list,clear,log,run,check,help,start,stop,restart,reload,status,enable,disable}
    list                Listing bots or queues
    clear               Clear a queue
    log                 Get last log lines of a bot
    run                 Run a bot interactively
    check               Check installation and configuration
    help                Show the help
    start               Start a bot or botnet
    stop                Stop a bot or botnet
    restart             Restart a bot or botnet
    reload              Reload a bot or botnet
    status              Status of a bot or botnet
    enable              Enable a bot
    disable             Disable a bot

        intelmqctl [start|stop|restart|status|reload] --group [collectors|parsers|experts|outputs]
        intelmqctl [start|stop|restart|status|reload] bot-id
        intelmqctl [start|stop|restart|status|reload]
        intelmqctl list [bots|queues|queues-and-status]
        intelmqctl log bot-id [number-of-lines [log-level]]
        intelmqctl run bot-id message [get|pop|send]
        intelmqctl run bot-id process [--msg|--dryrun]
        intelmqctl run bot-id console
        intelmqctl clear queue-id
        intelmqctl check
```

Intelmqctl should be used with the intelmq user.

Hackers use terminal!

As you probably discover now, there are 3 or 4 configurations files:

- runtime.conf > Configuration of the bots
- pipeline.conf > How bots are organized
- harmonization.conf > List of available fields
- modify.conf > One or multiple files for managing modify bot

2 others files help you when you need it the most:

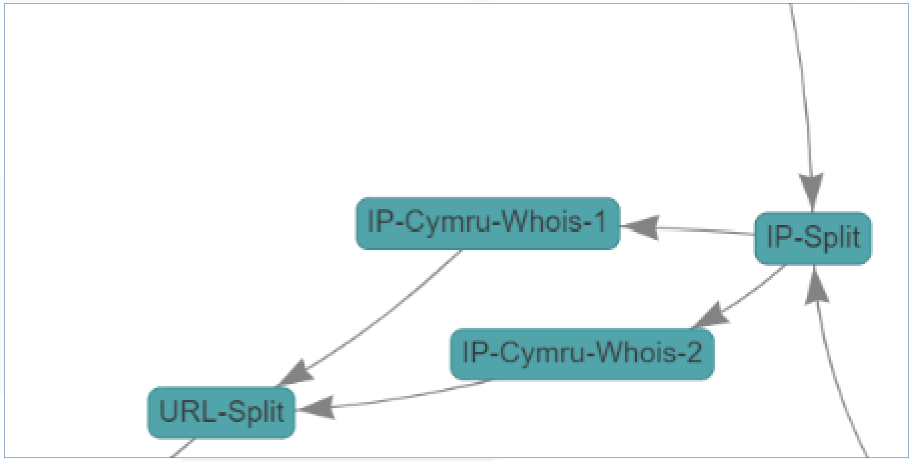
- BOTS > This file provides ave the skeleton on each bots
- feeds.yaml > This file provides you url of feeds, rate limiting...

Intelmqctl should be used with the intelmq user.

Advanced

If one of the bot takes too much time to process data, you can:

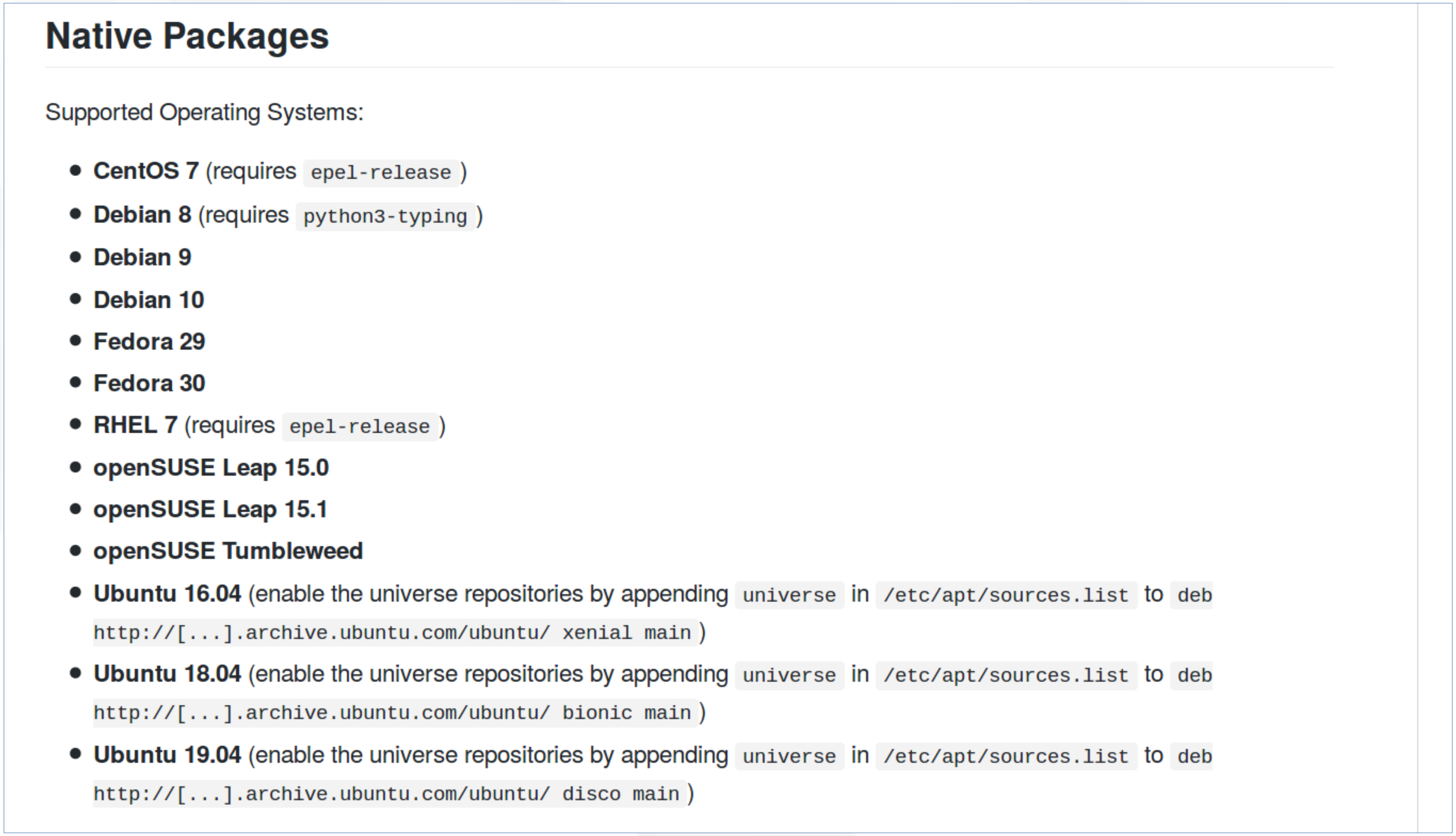
- Parallelize the process on the same server



- Or balancing the loads between multiple server!

Installation

IntelMQ is available from package managers, or for manual install (pip, git)



Native Packages

Supported Operating Systems:

- **CentOS 7** (requires `epel-release`)
- **Debian 8** (requires `python3-typing`)
- **Debian 9**
- **Debian 10**
- **Fedora 29**
- **Fedora 30**
- **RHEL 7** (requires `epel-release`)
- **openSUSE Leap 15.0**
- **openSUSE Leap 15.1**
- **openSUSE Tumbleweed**
- **Ubuntu 16.04** (enable the universe repositories by appending `universe` in `/etc/apt/sources.list` to `deb http://[...].archive.ubuntu.com/ubuntu/ xenial main`)
- **Ubuntu 18.04** (enable the universe repositories by appending `universe` in `/etc/apt/sources.list` to `deb http://[...].archive.ubuntu.com/ubuntu/ bionic main`)
- **Ubuntu 19.04** (enable the universe repositories by appending `universe` in `/etc/apt/sources.list` to `deb http://[...].archive.ubuntu.com/ubuntu/ disco main`)

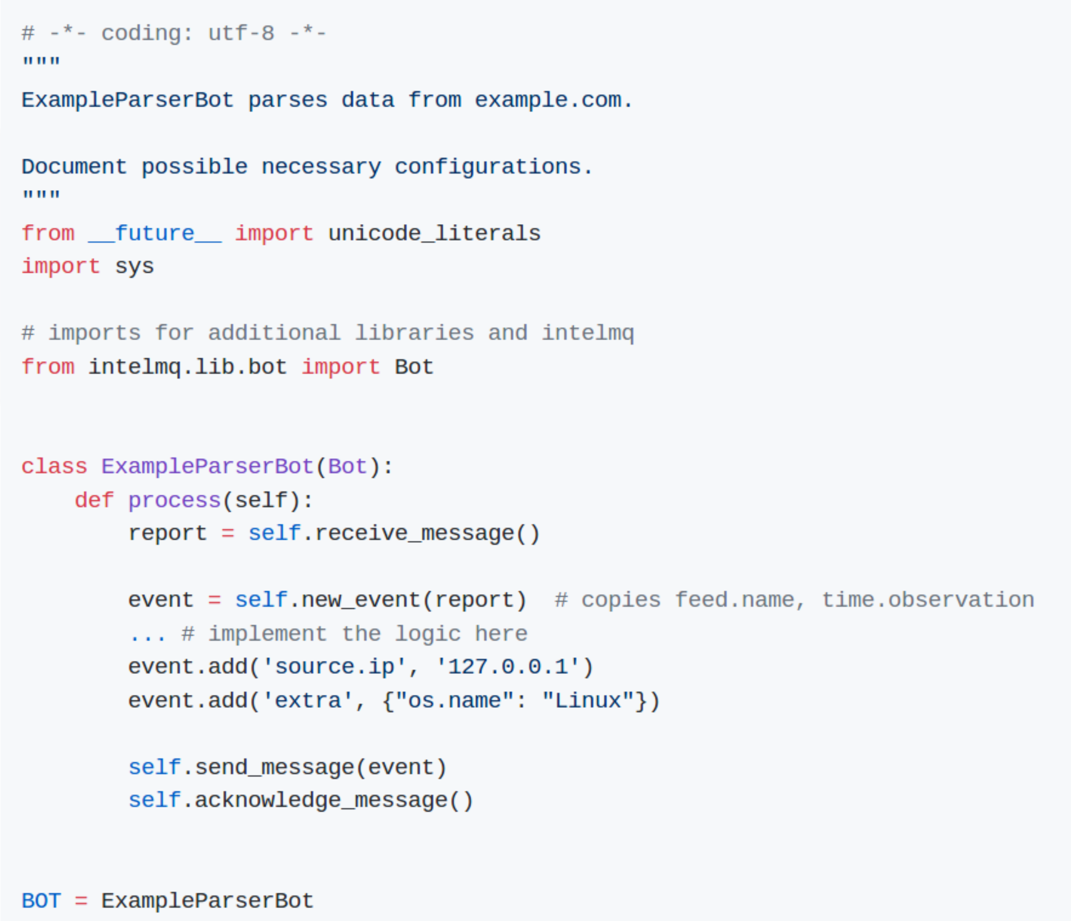
https://github.com/certtools/intelmq/blob/develop/docs/INSTALL.md

Development

Documentation available in the github is sufficient to start developing you own bot.

https://github.com/certtools/intelmq/blob/develop/docs/Developers-Guide.md

This guide help you for setting your machine, and provides a skeleton of bot:

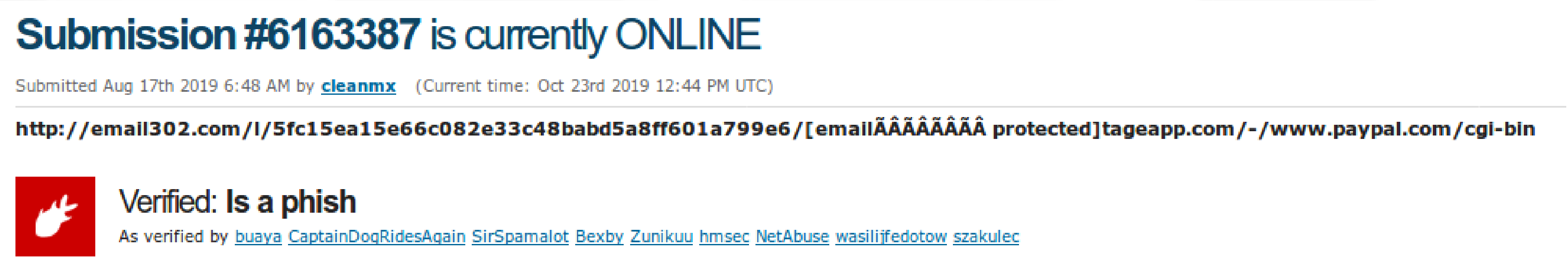


```
# -*- coding: utf-8 -*-
"""
ExampleParserBot parses data from example.com.

Document possible necessary configurations.
"""
from __future__ import unicode_literals
import sys

# imports for additional libraries and intelmq
from intelmq.lib.bot import Bot


class ExampleParserBot(Bot):
    def process(self):
        report = self.receive_message()

        event = self.new_event(report)  # copies feed.name, time.observation
        ... # implement the logic here
        event.add('source.ip', '127.0.0.1')
        event.add('extra', {"os.name": "Linux"})

        self.send_message(event)
        self.acknowledge_message()


BOT = ExampleParserBot
```

Last words

IntelMQ is a great tool, but unfortunately is not enough….

ALWAYS validate your IOC before using it!



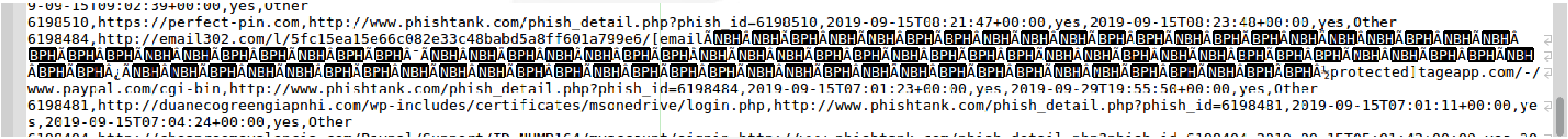
Submission #6163387 is currently ONLINE

Submitted Aug 17th 2019 6:48 AM by cleanmx   (Current time: Oct 23rd 2019 12:44 PM UTC)

http://email302.com/l/5fc15ea15e66c082e33c48babd5a8ff601a799e6/[emailÃÂÃÂÃÂÃÂÃ protected]tageapp.com/-/www.paypal.com/cgi-bin

Verified: **Is a phish**

As verified by buaya CaptainDogRidesAgain SirSpamalot Bexby Zunikuu hmsec NetAbuse wasilijfedotow szakulec



```
9-09-15T09:02:39+00:00,yes,Other
6198510,https://perfect-pin.com,http://www.phishtank.com/phish_detail.php?phish_id=6198510,2019-09-15T08:21:47+00:00,yes,2019-09-15T08:23:48+00:00,yes,Other
6198484,http://email302.com/l/5fc15ea15e66c082e33c48babd5a8ff601a799e6/[emailÃNBHÂNBHÃBPHÂNBHÃNBHÂBPHÃBPHÂNBHÃNBHÂNBHÃBPHÂBPHÃNBHÂBPHÃBPHÂNBHÃNBHÂNBHÃBPHÂNBHÃNBHÂ
BPHÃBPHÂBPHÃNBHÂNBHÃBPHÂBPHÃNBHÂBPHÃBPHÂ¯ÃNBHÂNBHÃBPHÂNBHÃNBHÂBPHÃBPHÂNBHÃNBHÂNBHÃBPHÂBPHÃNBHÂBPHÃBPHÂBPHÃNBHÂNBHÃBPHÂNBHÃNBHÂBPHÃBPHÂBPHÃNBHÂNBHÃBPHÂBPHÃNBH
ÂBPHÃBPHÂ¿ÃNBHÂNBHÃBPHÂNBHÃNBHÂBPHÃBPHÂNBHÃNBHÂNBHÃBPHÂBPHÃNBHÂBPHÃBPHÂBPHÃNBHÂNBHÃBPHÂNBHÃNBHÂBPHÃBPHÂBPHÃNBHÂNBHÃBPHÂBPHÃNBHÂBPHÃBPHÂ½protected]tageapp.com/-/
www.paypal.com/cgi-bin,http://www.phishtank.com/phish_detail.php?phish_id=6198484,2019-09-15T07:01:23+00:00,yes,2019-09-29T19:55:50+00:00,yes,Other
6198481,http://duanecogreengiapnhi.com/wp-includes/certificates/msonedrive/login.php,http://www.phishtank.com/phish_detail.php?phish_id=6198481,2019-09-15T07:01:11+00:00,ye
s,2019-09-15T07:04:24+00:00,yes,Other
```

Questions?

Celine Massompierre
@kalyparker

kalyparker@protonmail.com

References

Images:
https://www.fatcow.com/free-icons
https://imgbin.com/png/kwnr4usQ/malware-analysis-computer-virus-computer-icons-computer-security-png
https://iphone.mob.org/game/worms.html
https://nakedsecurity.sophos.com/2019/06/10/the-goldbrute-botnet-is-trying-to-crack-open-1-5-million-rdp-servers/
https://github.com/ustayready/CredSniper
https://fcw.com/articles/2015/09/16/malware-dns-haystack.aspx
https://wiki.pentaho.com/display/BAD/About+Kettle+and+Big+Data
https://mentalfloss.com/article/92127/how-many-combinations-are-possible-using-6-lego-bricks
https://www.tibco.com/blog/2015/11/30/incremental-composition-the-engine-of-agility/

IntelMQ
https://github.com/certtools/intelmq/
https://www.enisa.europa.eu/topics/csirt-cert-services/community-projects/incident-handling-automation
https://www.trusted-introducer.org/Incident-Classification-Taxonomy.pdf

IntelMQ-Manager
https://github.com/certtools/intelmq-manager