enisa

EUROPEAN UNION AGENCY
FOR CYBERSECURITY

ORCHESTRATION OF CSIRT TOOLS

Training Slides
Technology Background

ENISA (Christian Van Heurck) and NASK

15 | 04 | 2021

CHAPTERS

- CHAPTER 1: MISP Administration Module
- CHAPTER 2: IntelMQ Administration Module
- CHAPTER 3: TheHive & Cortex Administration Module
- CHAPTER 4: MISP Analyst Module
- CHAPTER 5: IntelMQ Analyst Module
- CHAPTER 6: TheHive & Cortex Analyst Module
- CHAPTER 7: Technology Background

CHAPTER 1

ORCHESTRATION OF CSIRT TOOLS

MISP ADMINISTRATION MODULE

MISP Administration Module

Introduction

Preconfigured states

For the exercise purposes, we prepared two states of the exercise that you can install by instructions provided in the next slides.

NOTE: More detailed instructions about all topics discussed in this presentation can be found in the student’s handbook. Please open them now.

Misp bare

This state consists of two MISP systems.

One (*https://misp.enisa.ex*) is not configured at all.

This represents state just after installation.

One account is available with username: **admin@admin.test** and password **admin**

Another instance (*https://misp2.enisa.ex*) contains data and minimal configuration.

Credentials: **admin@admin.test**:**SecondInstancePassword123!**

API Key gxPEOFh04jGZriMUhBI3U9IyOp7lrxKYifIDMMB3

Misp configured

This represents both misps in configured condition.

Configured by the steps provided in the student’s handbook.

Configured state contains some random events, so you can look at them and click around.

MISP Administration Module

Exercise

Installation

Let’s start with setting basic configuration options.

- *cd /opt/enisa/trainings-2019/admin/misp*
- To start the exercise type in *./start-exercise.sh*
- Navigate to your organization's MISP with web browser (*https://misp.enisa.ex*)
- Login with **admin@admin.test**:**admin**
- Change password to **Str0ngP@sswd!**
- Set baseurl to *https://misp.enisa.ex*
- Edit existing organisation
- Make MISP alive!

Events

Events are the core of misp instance.

They allow you to manage, share and enrich intelligence of yours and others organisations.

- Add an event in **Event Actions** -> **Add Event**
- List events with **Event Actions** -> **List Events**

Galaxies

In MISP, galaxies are used to express a large object called cluster.

They are formed by elements (key:value pairs). Default vocabularies are available in MISP galaxy – they can be overwritten, replaced or updated.

- Enable and Update galaxies with **Galaxies** -> **Update Galaxies**
  NOTE: Updating galaxies is only possible with internet access.
- Check what you can do with galaxies on your event
- To add galaxy to the event go to event view and click **Galaxies -> Add**

Taxonomies

Taxonomy is a group of „machine tags” used to tag events and attributes.

Every tag is composed of a namespace (mandatory), a predicate (mandatory) and a value (optional).

Example: osint:source-type=”blog-post” (osint - namespace, source-type - predicate, “blog-post” - value).

- These machine tags are often called **triple tag** due to their format.
- To enable default taxonomies, click on **Event Actions** -> **List Taxonomies** -> **Update Taxonomies**

  NOTE: Updating galaxies is only possible with internet access.
- Enable taxonomies in Taxonomies View
- Add taxonomy to your event

User management

Adding new user is very simple.

To add new user go to **Administration** -> **Add User**

You need to fill following fields:

- **Email** - email of the user.
- **Organisation** - choose accordingly depending on which organisation user belongs to.
- **Role** - this determines what user can do in the misp instance. Read the next section for quick overview of permission system.
- Click *Submit*

Organisations

Each users belongs to an organisation. As admin, you can manage these organisations.

Organisations allow for separation of users and synchronisation process.

- To add new organisation click on the **Add Organisation** button in the administration menu
- To list all organisations click **List Organisations** under the administration menu to the left

Role permissions

Role Permission system in MISP consists of many permissions.

MISP user roles can be found under **Global Actions** -> **Role Permissions**.

- List of all role permissions and predefined roles in MISP can be found in the student’s handbook
- Read up on them!

Dashboard and Statistics

Dashboard and statistics allow to monitor state of the MISP instance.

Dashboard can be found in the **Global Actions** -> **Dashboard.**

Statistics are located under **Global Actions** -> **Statistics.**

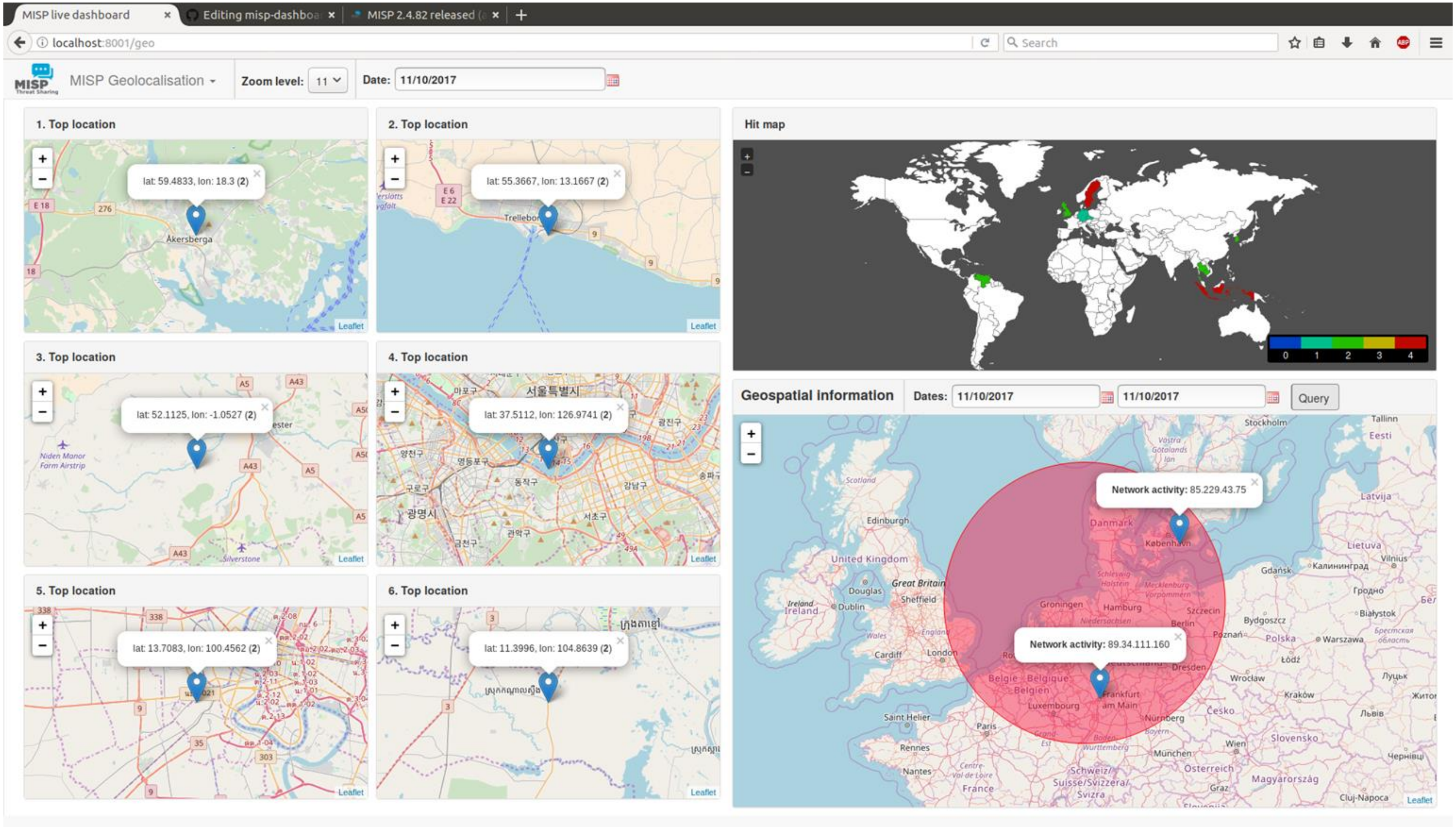
Dashboard and Statistics

Dashboard and statistics allow to monitor state of the MISP instance.

Dashboard can be found in the **Global Actions** -> **Dashboard.**

Statistics are located under **Global Actions** -> **Statistics.**

Dashboard



Zoom level: 11

Date: 11/10/2017

1. Top location

lat: 59.4833, lon: 18.3 (2)

2. Top location

lat: 55.3667, lon: 13.1667 (2)

3. Top location

lat: 52.1125, lon: -1.0527 (2)

4. Top location

lat: 37.5112, lon: 126.9741 (2)

5. Top location

lat: 13.7083, lon: 100.4562 (2)

6. Top location

lat: 11.3996, lon: 104.8639 (2)

Hit map

Geospatial information

Dates: 11/10/2017 11/10/2017

Network activity: 85.229.43.75

Network activity: 89.34.111.160

Automation API

Automation allows for automating tasks using MISP API.

Automation options can be found in **Event Actions** -> **Automation tab**.

Inside the **Automation tab** you can find *API key* as well as list of endpoints that *MISP API* exposes.

This topic is very complex, you can read up on the topic at *https://www.circl.lu/doc/misp/automation/#automation-api*.

Synchronisation

Synchronisation allows to exchange data between instances.

This can improve cooperation between organisations and allow for easy *IoC/data* exchange.

Common way of synchronizing the *MISPs* is as follows:

- Add **OrgB** as a local organisation on **ServerA (OrgB**.**ServerA**) using **OrgB's** existing *UUID* from their local organisation on **ServerB**.
- Add a Sync User (syncuser@**OrgB**.**ServerA**) in the organisation **OrgB**.**ServerA** on the *MISP* **ServerA**.
- Set up a sync server on *MISP* **ServerB** using the key (called *Authkey*) from the sync user (*syncuser@***OrgB**.**ServerA**) created on **MISP ServerA**.

CHAPTER 2

ORCHESTRATION OF CSIRT TOOLS INTELMQ ADMINISTRATION MODULE

IntelMQ Administration Module

Introduction

IntelMQ - introduction

A system for incident response teams to collect, process and analyze data from various sources using a message queue protocol.

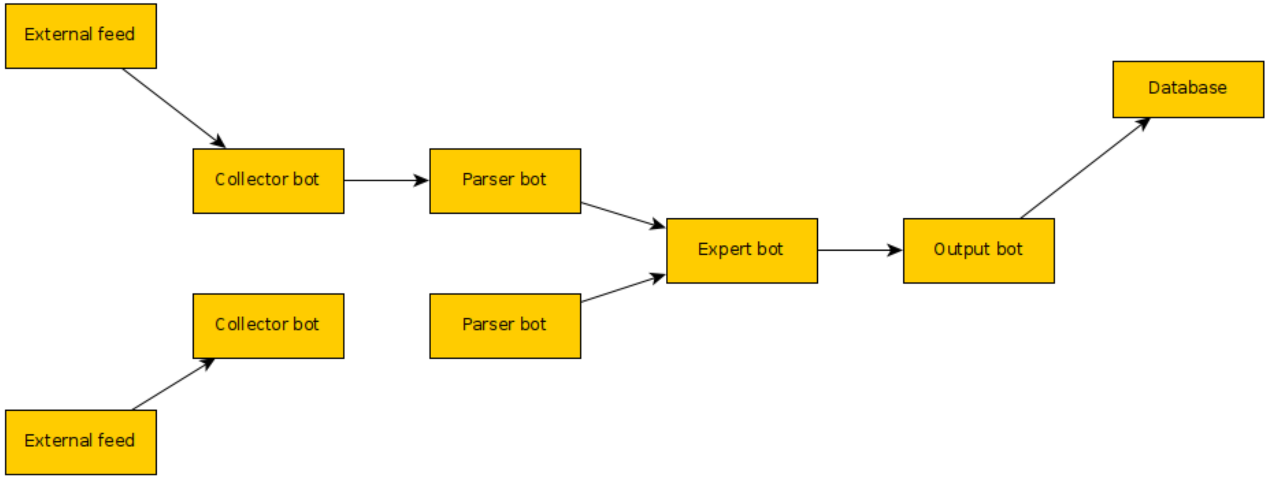
IntelMQ - bots

Four kind of bot nodes:

- Collectors - used to collect data
- Parsers - used to parse raw data
- Experts - used to process and enrich the existing data
- Output - exit nodes that allow us to save the result of the whole

IntelMQ - pipeline

Bots may be connected to create a **pipeline**



Task 1 – simple pipeline

Run clean installation of IntelMQ

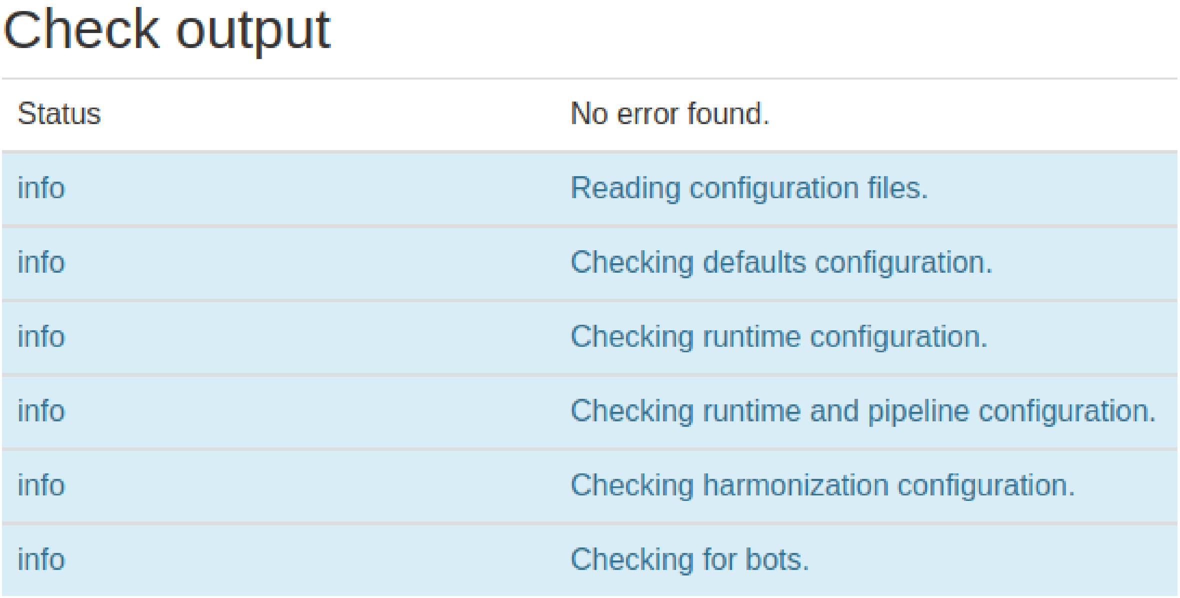
```
$ cd /opt/enisa/trainings-2019/admin/intelmq/intelmq-clean
$ ./start_exercise.sh
```

IntelMQ Administration Module

Exercises

Task 1 – simple pipeline

Check if everything is ok on “Check” tab at http://intelmq.enisa.ex



Check output

Status	No error found.
info	Reading configuration files.
info	Checking defaults configuration.
info	Checking runtime configuration.
info	Checking runtime and pipeline configuration.
info	Checking harmonization configuration.
info	Checking for bots.

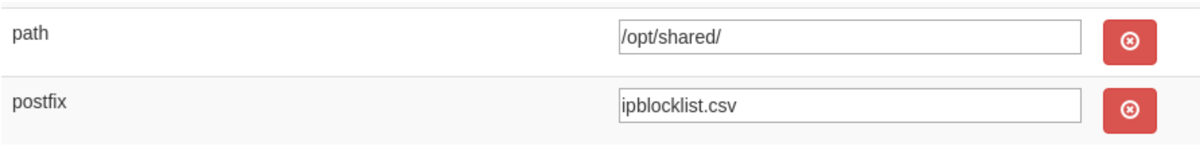
Task 1 – simple pipeline

Configure collector

1. Choose configuration tab

2. Press “Add bot” button and place it anywhere on the board. From menu to the left choose Collector -> File

3. Put /opt/shared/ipblocklist.csv path in node config like shown below:



path	/opt/shared/	⊗
postfix	ipblocklist.csv	⊗

4. Name the feed and data provider (fields “name” and “provider”) with custom, descriptive name. It will be useful in pipelines with more feeds to see the source and type of data in the output.

5. Press OK to add the bot

Task 1 – simple pipeline

Configure output

1. Create output node and place it on the board. As the type choose “File”

2. Configure it to output data to a temporary file at /opt/shared/out - this file will be visible in VM under `/opt/enisa/trainings-2019/admin/intelmq/intelmq-clean/shared/out`

Make sure that file is world-writable:

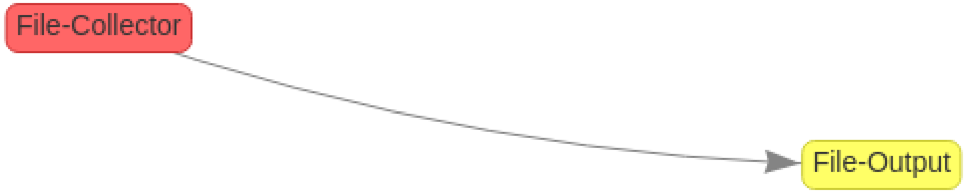
```
$ chmod 666 /opt/enisa/trainings-2019/admin/intelmq/intelmq-clean/shared/out
```

Task 1 – simple pipeline

Make the connection between collector and output

1. Press “Add queue” button

2. Create the connection



Remember to always press **Save configuration** button after making any changes!

Task 2 – test pipeline

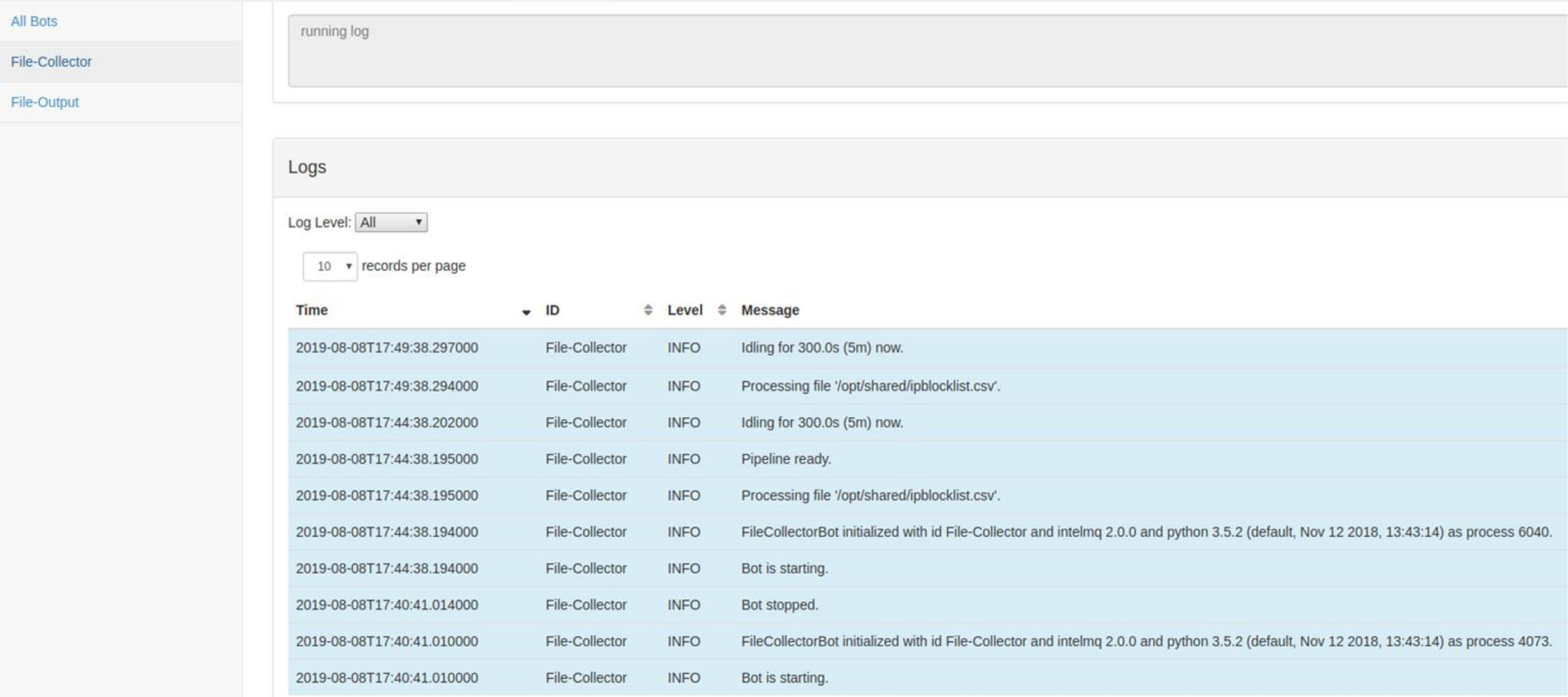
Make the connection between collector and output

1. Choose management tab
2. Run pipeline under “Whole Botnet Status”
3. Check if output file is being populated

```
$ cat /opt/enisa/trainings-2019/admin/intelmq/intelmq-clean/shared/out
```

Task 2 – test pipeline

Check “Monitor” tab to see bots logs



Logs

Log Level: All

10 records per page

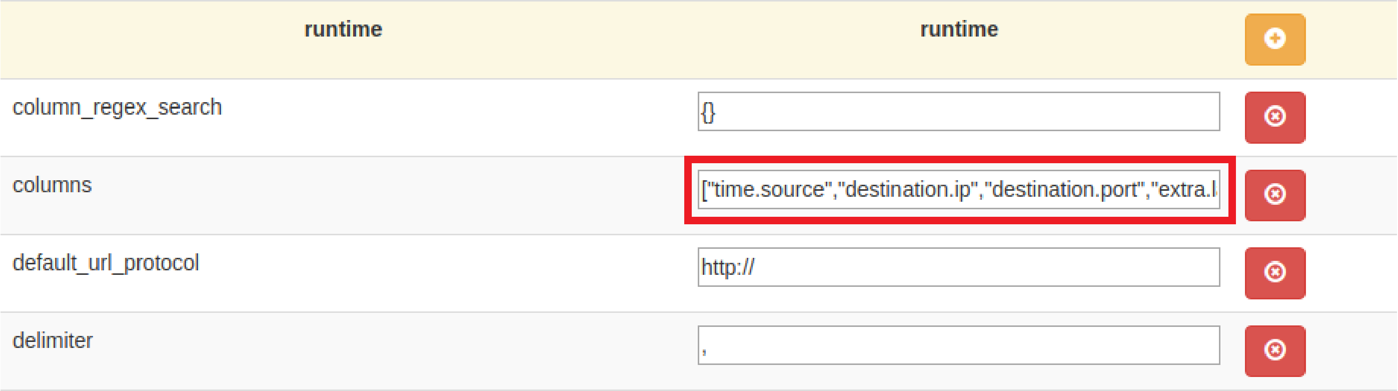
Time	ID	Level	Message
2019-08-08T17:49:38.297000	File-Collector	INFO	Idling for 300.0s (5m) now.
2019-08-08T17:49:38.294000	File-Collector	INFO	Processing file '/opt/shared/ipblocklist.csv'.
2019-08-08T17:44:38.202000	File-Collector	INFO	Idling for 300.0s (5m) now.
2019-08-08T17:44:38.195000	File-Collector	INFO	Pipeline ready.
2019-08-08T17:44:38.195000	File-Collector	INFO	Processing file '/opt/shared/ipblocklist.csv'.
2019-08-08T17:44:38.194000	File-Collector	INFO	FileCollectorBot initialized with id File-Collector and intelmq 2.0.0 and python 3.5.2 (default, Nov 12 2018, 13:43:14) as process 6040.
2019-08-08T17:44:38.194000	File-Collector	INFO	Bot is starting.
2019-08-08T17:40:41.014000	File-Collector	INFO	Bot stopped.
2019-08-08T17:40:41.010000	File-Collector	INFO	FileCollectorBot initialized with id File-Collector and intelmq 2.0.0 and python 3.5.2 (default, Nov 12 2018, 13:43:14) as process 4073.
2019-08-08T17:40:41.010000	File-Collector	INFO	Bot is starting.

Task 3 – Add parser and expert bots

Add and configure parser and expert

1. Add Generic CSV parser
2. Configure “columns” field like shown below:

["time.source","destination.ip","destination.port","extra.lastOnline","classification.identifier"]



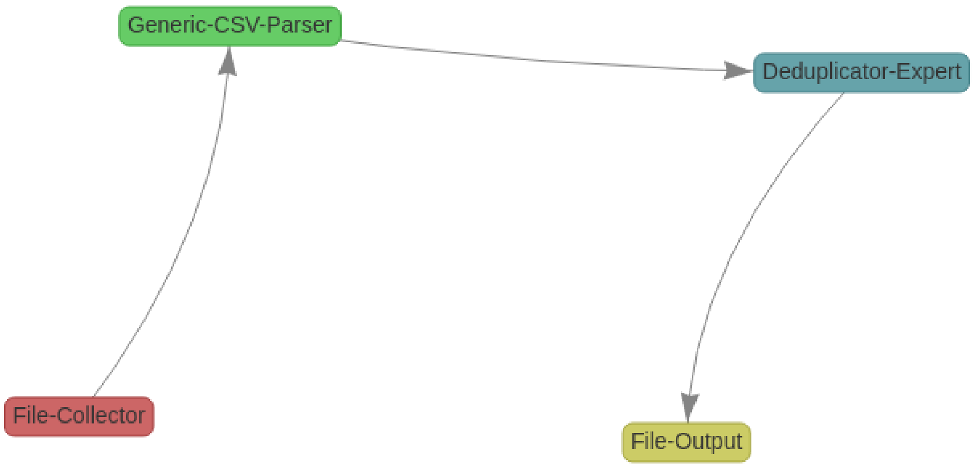
runtime	runtime	⊕
column_regex_search	{}	⊗
columns	["time.source","destination.ip","destination.port","extra.l	⊗
default_url_protocol	http://	⊗
delimiter	,	⊗

3. Add deduplica tor expert. Leave its configuration as default.

Task 3 – Add parser and expert bots

Configure connections

Create connections between nodes to look like shown below

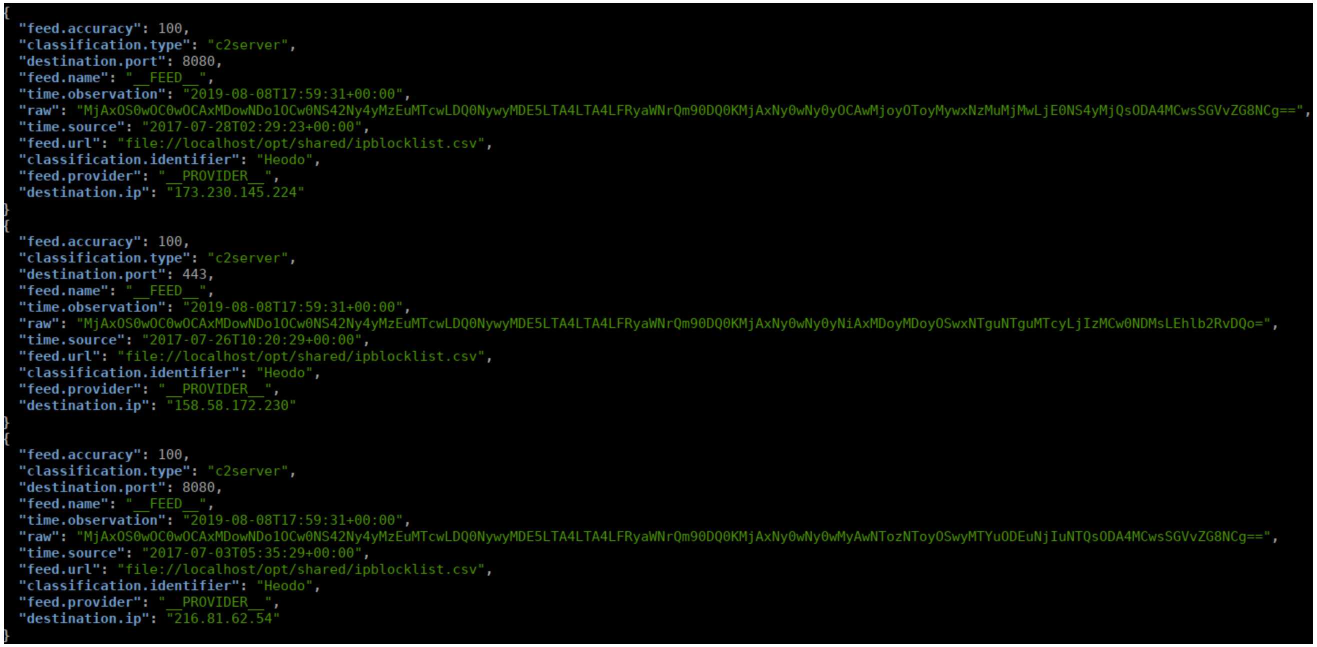


Task 3 – Add parser and expert bots

Test pipeline

Stop the pipeline, clear output file and rerun pipeline. Now the output file should look like this

```
$ cat /opt/enisa/trainings-2019/admin/intelmq/intelmq-clean/shared/out | jq
```



```
{
  "feed.accuracy": 100,
  "classification.type": "c2server",
  "destination.port": 8080,
  "feed.name": "__FEED__",
  "time.observation": "2019-08-08T17:59:31+00:00",
  "raw": "MjAxOS0wOC0wOCAxMDowNDo1OCw0NS42Ny4yMzEuMTcwLDQ0NywyMDE5LTA4LTA4LFRyaWNrQm90DQ0KMjAxNy0wNy0yOCAwMjoyOToyMywxNzMuMjMwLjE0NS4yMjQsODA4MCwsSGVvZG8NCg==",
  "time.source": "2017-07-28T02:29:23+00:00",
  "feed.url": "file://localhost/opt/shared/ipblocklist.csv",
  "classification.identifier": "Heodo",
  "feed.provider": "__PROVIDER__",
  "destination.ip": "173.230.145.224"
}
{
  "feed.accuracy": 100,
  "classification.type": "c2server",
  "destination.port": 443,
  "feed.name": "__FEED__",
  "time.observation": "2019-08-08T17:59:31+00:00",
  "raw": "MjAxOS0wOC0wOCAxMDowNDo1OCw0NS42Ny4yMzEuMTcwLDQ0NywyMDE5LTA4LTA4LFRyaWNrQm90DQ0KMjAxNy0wNy0yNiAxMDoyMDoyOSwxNTguNTguMTcyLjIzMCw0NDMsLEhlb2RvDQo=",
  "time.source": "2017-07-26T10:20:29+00:00",
  "feed.url": "file://localhost/opt/shared/ipblocklist.csv",
  "classification.identifier": "Heodo",
  "feed.provider": "__PROVIDER__",
  "destination.ip": "158.58.172.230"
}
{
  "feed.accuracy": 100,
  "classification.type": "c2server",
  "destination.port": 8080,
  "feed.name": "__FEED__",
  "time.observation": "2019-08-08T17:59:31+00:00",
  "raw": "MjAxOS0wOC0wOCAxMDowNDo1OCw0NS42Ny4yMzEuMTcwLDQ0NywyMDE5LTA4LTA4LFRyaWNrQm90DQ0KMjAxNy0wNy0wMyAwNTozNToyOSwyMTYuODEuNjIuNTQsODA4MCwsSGVvZG8NCg==",
  "time.source": "2017-07-03T05:35:29+00:00",
  "feed.url": "file://localhost/opt/shared/ipblocklist.csv",
  "classification.identifier": "Heodo",
  "feed.provider": "__PROVIDER__",
  "destination.ip": "216.81.62.54"
}
```

Task 4 – Use more complex collector and output bots.

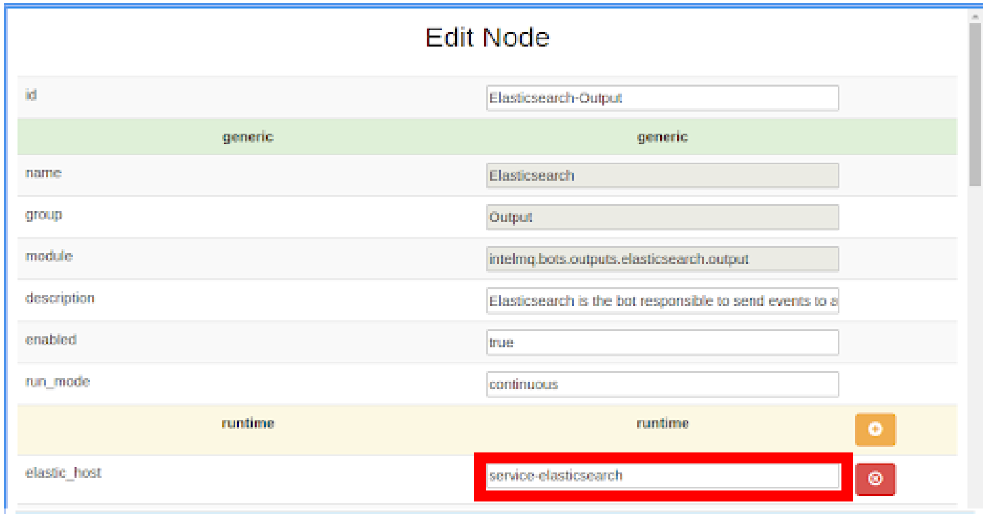
Run script sending malicious requests

```
$ cd /opt/enisa/trainings-2019/admin/intelmq/scripts
$ python3 send.py honeypot.enisa.ex
```

Task 4 – Use more complex collector and output bots.

Add new nodes

1. Add file input bot. As the input file put the path /opt/shared/snare.log (remember to name feed and provider correctly!)
2. As the parser bot use SNARE - our customly created one
3. Add deduplicator, just like in previous task
3. As the output we'll use Elasticsearch. Choose Elasticsearch output bot and configure it to have "elastic_host" option as "service-elasticsearch"



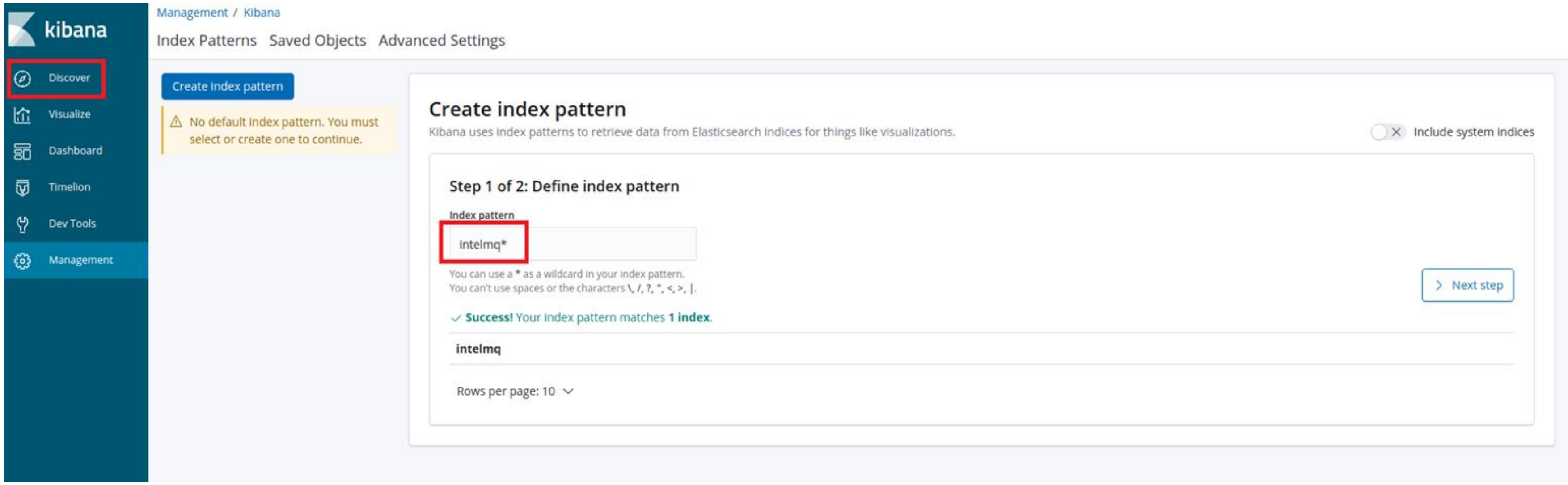
Edit Node

id	Elasticsearch-Output
generic	generic
name	Elasticsearch
group	Output
module	intelmq.bots.outputs.elasticsearch.output
description	Elasticsearch is the bot responsible to send events to a
enabled	true
run_mode	continuous
runtime	runtime
elastic_host	service-elasticsearch

Task 4 – Use more complex collector and output bots.

Test pipeline

1. Visit kibana.enisa.ex
2. Click “Discover” tab and create index pattern named “intelmq”



No default index pattern. You must select or create one to continue.

Create index pattern

Kibana uses index patterns to retrieve data from Elasticsearch indices for things like visualizations.

Step 1 of 2: Define index pattern

Index pattern: intelmq*

You can use a * as a wildcard in your index pattern.
You can't use spaces or the characters \, /, ?, ", <, >, |.

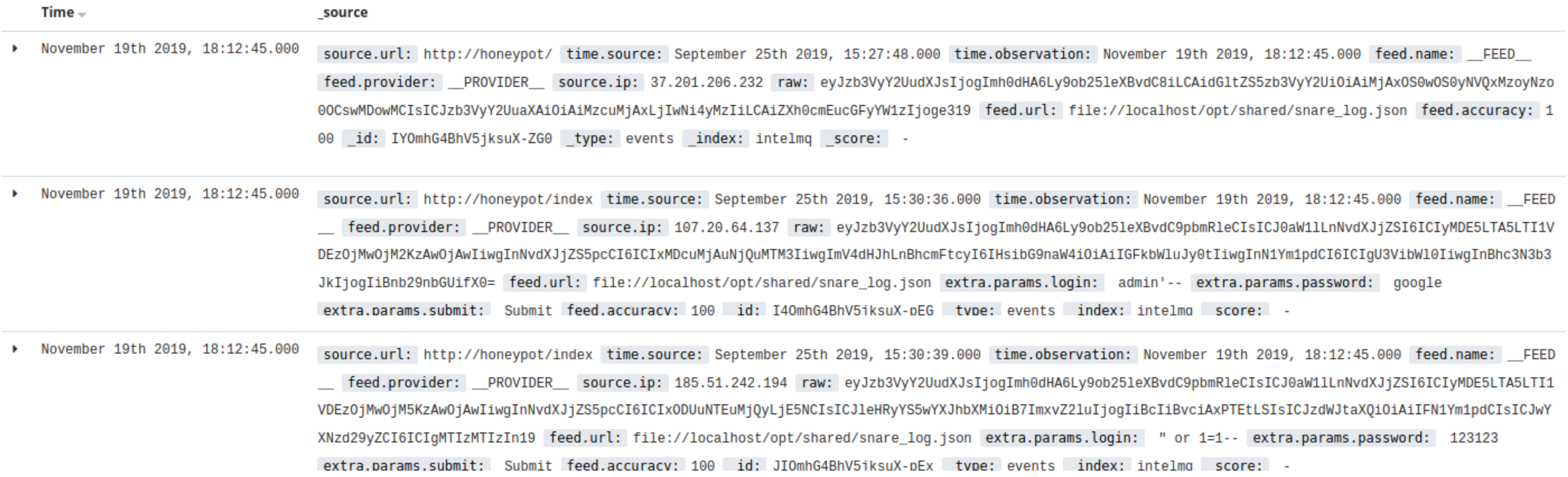
✓ **Success!** Your index pattern matches **1 index.**

intelmq

Task 4 – Use more complex collector and output bots.

Test pipeline

1. Press next
2. In “Time Filter field name” put “time.observation” and press “Create index pattern”
3. If everything went well, you should be able to see your data and easily use the search features provided by ES:



```
Time                                  _source

▸ November 19th 2019, 18:12:45.000    source.url: http://honeypot/ time.source: September 25th 2019, 15:27:48.000 time.observation: November 19th 2019, 18:12:45.000 feed.name: __FEED__
                                      feed.provider: __PROVIDER__ source.ip: 37.201.206.232 raw: eyJzb3VyY2UudXJsIjogImh0dHA6Ly9ob25leXBvdC8iLCAidGltZS5zb3VyY2UiOiAiMjAxOS0wOS0yNVQxMzoyNzo
                                      0OCswMDowMCIsICJzb3VyY2UuaXAiOiAiMzcuMjAxLjIwNi4yMzIiLCAiZXh0cmEucGFyYW1zIjoge319 feed.url: file://localhost/opt/shared/snare_log.json feed.accuracy: 1
                                      00 _id: IYOmhG4BhV5jksuX-ZG0 _type: events _index: intelmq _score: -

▸ November 19th 2019, 18:12:45.000    source.url: http://honeypot/index time.source: September 25th 2019, 15:30:36.000 time.observation: November 19th 2019, 18:12:45.000 feed.name: __FEED
                                      __ feed.provider: __PROVIDER__ source.ip: 107.20.64.137 raw: eyJzb3VyY2UudXJsIjogImh0dHA6Ly9ob25leXBvdC9pbmRleCIsICJ0aW1lLnNvdXJjZSI6ICIyMDE5LTA5LTI1V
                                      DEzOjMwOjM2KzAwOjAwIiwgInNvdXJjZS5pcCI6ICIxMDcuMjAuNjQuMTM3IiwgImV4dHJhLnBhcmFtcyI6IHsibG9naW4iOiAiIGFkbWluJy0tIiwgInN1Ym1pdCI6ICIgU3VibWl0IiwgInBhc3N3b3
                                      JkIjogIiBnb29nbGUifX0= feed.url: file://localhost/opt/shared/snare_log.json extra.params.login:  admin'-- extra.params.password:  google
                                      extra.params.submit:  Submit feed.accuracy: 100 _id: I4OmhG4BhV5jksuX-pEG _type: events _index: intelmq _score: -

▸ November 19th 2019, 18:12:45.000    source.url: http://honeypot/index time.source: September 25th 2019, 15:30:39.000 time.observation: November 19th 2019, 18:12:45.000 feed.name: __FEED
                                      __ feed.provider: __PROVIDER__ source.ip: 185.51.242.194 raw: eyJzb3VyY2UudXJsIjogImh0dHA6Ly9ob25leXBvdC9pbmRleCIsICJ0aW1lLnNvdXJjZSI6ICIyMDE5LTA5LTI1
                                      VDEzOjMwOjM5KzAwOjAwIiwgInNvdXJjZS5pcCI6ICIxODUuNTEuMjQyLjE5NCIsICJleHRyYS5wYXJhbXMiOiB7ImxvZ2luIjogIiBcIiBvciAxPTEtLSIsICJzdWJtaXQiOiAiIFN1Ym1pdCIsICJwY
                                      XNzd29yZCI6ICIgMTIzMTIzIn19 feed.url: file://localhost/opt/shared/snare_log.json extra.params.login:  " or 1=1-- extra.params.password:  123123
                                      extra.params.submit:  Submit feed.accuracy: 100 _id: JIOmhG4BhV5jksuX-pEx _type: events _index: intelmq _score: -
```

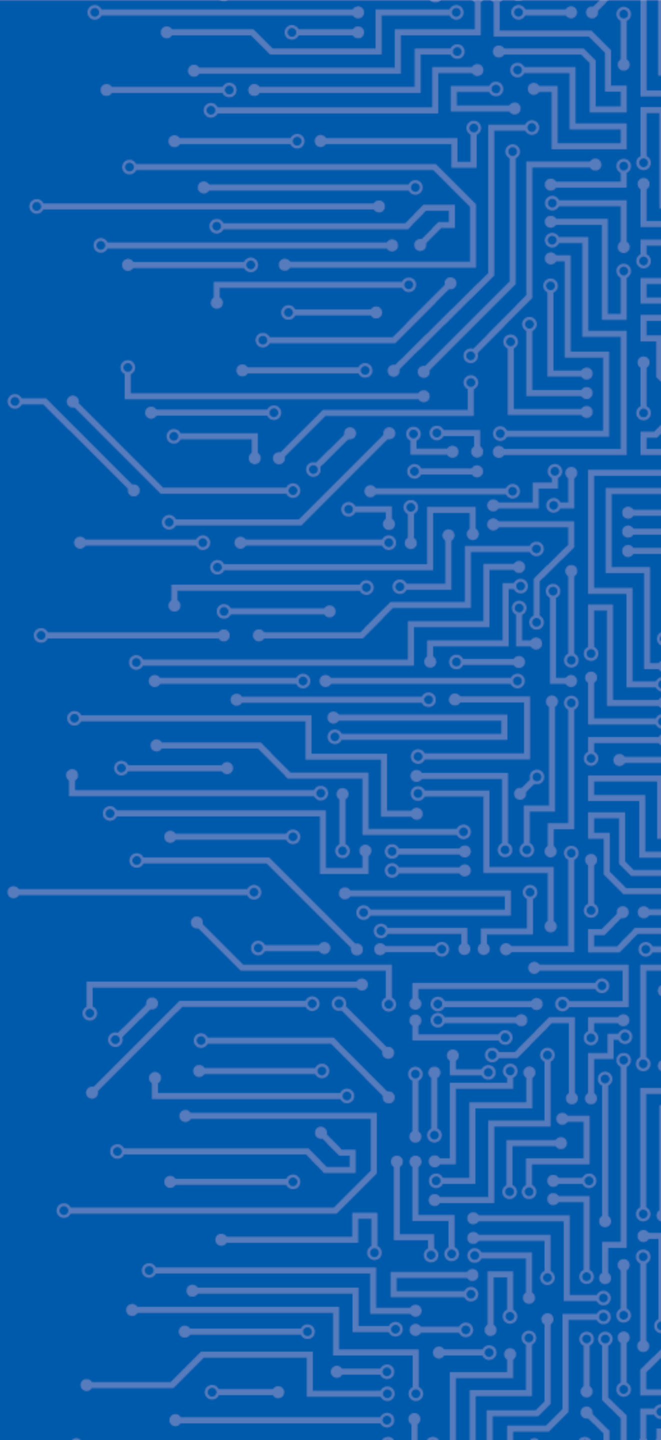
CHAPTER 3

ORCHESTRATION OF CSIRT TOOLS THEHIVE & CORTEX ADMINISTRATION MODULE

AGENDA

- Introduction to exercise
- Task 1: Setup TheHive & Cortex accounts
- Task 2: Configure Cortex analyzers
- Task 3: Configure the Hive-Cortex integration
- Task 4: Configure the Hive-MISP integration
- Task 5: Creating custom Cortex analyzer
- Task 6: Report templates, Case templates, Dashboards

TheHive & Cortex
Administration Module



Introduction

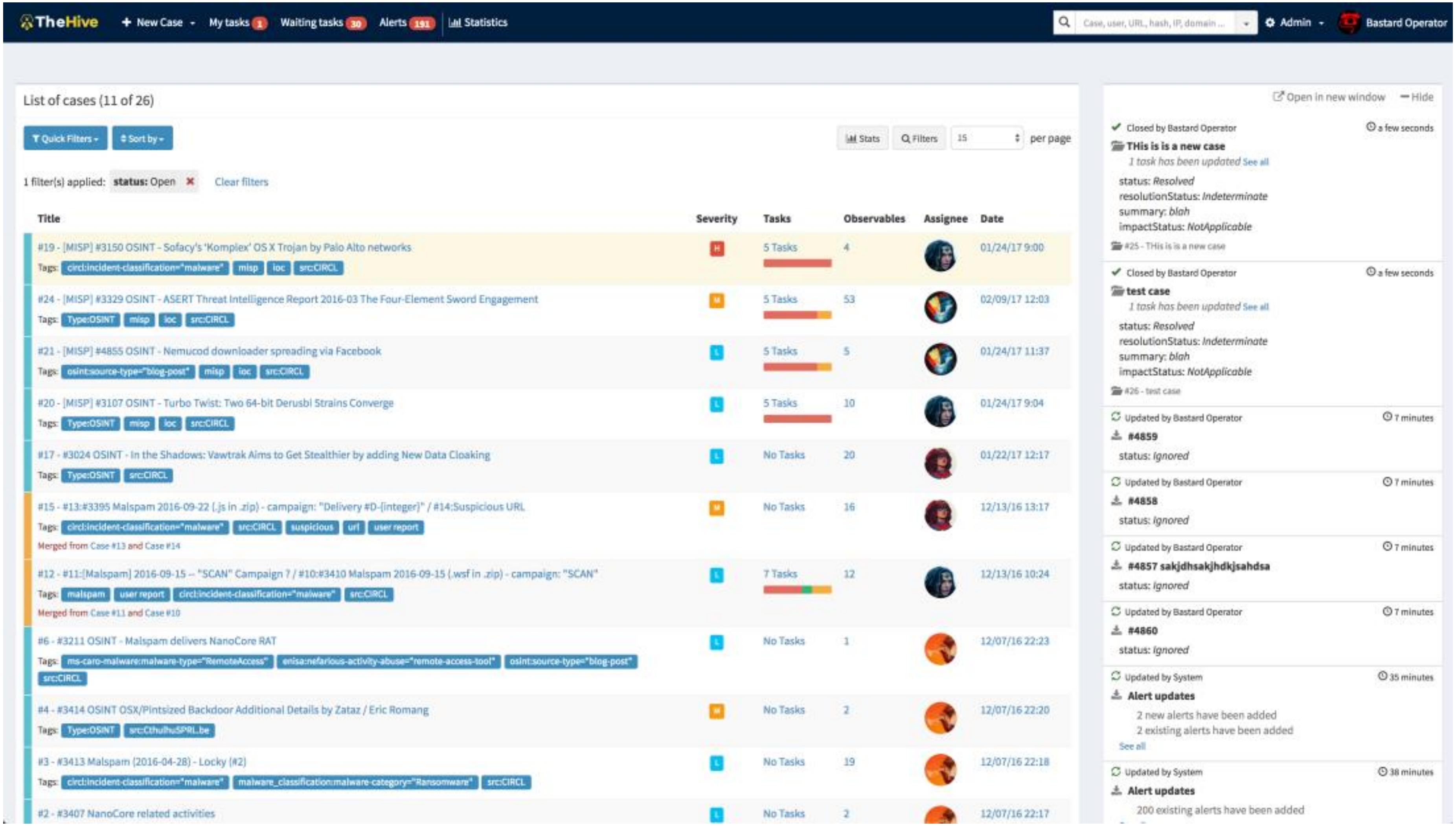
Why TheHive?

- System is dedicated for Security Operational Centers
- Easy way to conduct investigations
- Many users can work in parallel
- Useful built-in tools for data enrichment
- Autocorrelation of tags and observables
- Noncomplicated integration with MISP

TheHive Alternatives

- **Maltego (Commercial)**
  - https://www.paterva.com/buy/maltego-clients.php
- **FIR - Fast Incident Response**
  - https://github.com/certsocietegenerale/FIR

TheHive main view



List of cases (11 of 26)

1 filter(s) applied: **status:** Open ✖ Clear filters

Title	Severity	Tasks	Observables	Assignee	Date
#19 - [MISP] #3150 OSINT - Sofacy's 'Komplex' OS X Trojan by Palo Alto networks Tags: circl:incident-classification="malware" misp ioc src:CIRCL	H	5 Tasks	4		01/24/17 9:00
#24 - [MISP] #3329 OSINT - ASERT Threat Intelligence Report 2016-03 The Four-Element Sword Engagement Tags: Type:OSINT misp ioc src:CIRCL	M	5 Tasks	53		02/09/17 12:03
#21 - [MISP] #4855 OSINT - Nemucod downloader spreading via Facebook Tags: osint:source-type="blog-post" misp ioc src:CIRCL	L	5 Tasks	5		01/24/17 11:37
#20 - [MISP] #3107 OSINT - Turbo Twist: Two 64-bit Derusbi Strains Converge Tags: Type:OSINT misp ioc src:CIRCL	L	5 Tasks	10		01/24/17 9:04
#17 - #3024 OSINT - In the Shadows: Vawtrak Aims to Get Stealthier by adding New Data Cloaking Tags: Type:OSINT src:CIRCL	L	No Tasks	20		01/22/17 12:17
#15 - #13:#3395 Malspam 2016-09-22 (.js in .zip) - campaign: "Delivery #D-(integer)" / #14:Suspicious URL Tags: circl:incident-classification="malware" src:CIRCL suspicious url user report Merged from Case #13 and Case #14	M	No Tasks	16		12/13/16 13:17
#12 - #11:[Malspam] 2016-09-15 -- "SCAN" Campaign ? / #10:#3410 Malspam 2016-09-15 (.wsf in .zip) - campaign: "SCAN" Tags: malspam user report circl:incident-classification="malware" src:CIRCL Merged from Case #11 and Case #10	L	7 Tasks	12		12/13/16 10:24
#6 - #3211 OSINT - Malspam delivers NanoCore RAT Tags: ms-caro-malware:malware-type="RemoteAccess" enisa:nefarious-activity-abuse="remote-access-tool" osint:source-type="blog-post" src:CIRCL	L	No Tasks	1		12/07/16 22:23
#4 - #3414 OSINT OSX/Pintsized Backdoor Additional Details by Zataz / Eric Romang Tags: Type:OSINT src:CthulhuSPRL.be	M	No Tasks	2		12/07/16 22:20
#3 - #3413 Malspam (2016-04-28) - Locky (#2) Tags: circl:incident-classification="malware" malware_classification:malware-category="Ransomware" src:CIRCL	L	No Tasks	19		12/07/16 22:18
#2 - #3407 NanoCore related activities	L	No Tasks	2		12/07/16 22:17

✓ Closed by Bastard Operator — a few seconds
THis is is a new case
1 task has been updated See all
status: *Resolved*
resolutionStatus: *Indeterminate*
summary: *blah*
impactStatus: *NotApplicable*
#25 - THis is is a new case

✓ Closed by Bastard Operator — a few seconds
test case
1 task has been updated See all
status: *Resolved*
resolutionStatus: *Indeterminate*
summary: *blah*
impactStatus: *NotApplicable*
#26 - test case

Updated by Bastard Operator — 7 minutes
#4859
status: *Ignored*

Updated by Bastard Operator — 7 minutes
#4858
status: *Ignored*

Updated by Bastard Operator — 7 minutes
#4857 sakjdhsakjhdkjsahdsa
status: *Ignored*

Updated by Bastard Operator — 7 minutes
#4860
status: *Ignored*

Updated by System — 35 minutes
Alert updates
2 new alerts have been added
2 existing alerts have been added
See all

Updated by System — 38 minutes
Alert updates
200 existing alerts have been added

Basic concepts

- **Case - root object of investigation**
- **Task - belongs to Case**
- **Observables - added during the investigation, similar to MISP attributes, can be marked as Indicators of Compromise**
- **Alerts - events can be imported eg. from MISP**

Basic concepts

Each observable must have:

- **TLP**
- **Tag, Description - or both**

Observables can be:

- **domain, IP, hash, file, url ... etc**
- **Flagged as IoC**
- **Tagged**
- **Exported as: csv, text or MISP compatible format**
- **Analyzed via Cortex Analyzers**
- **exported to MISP**

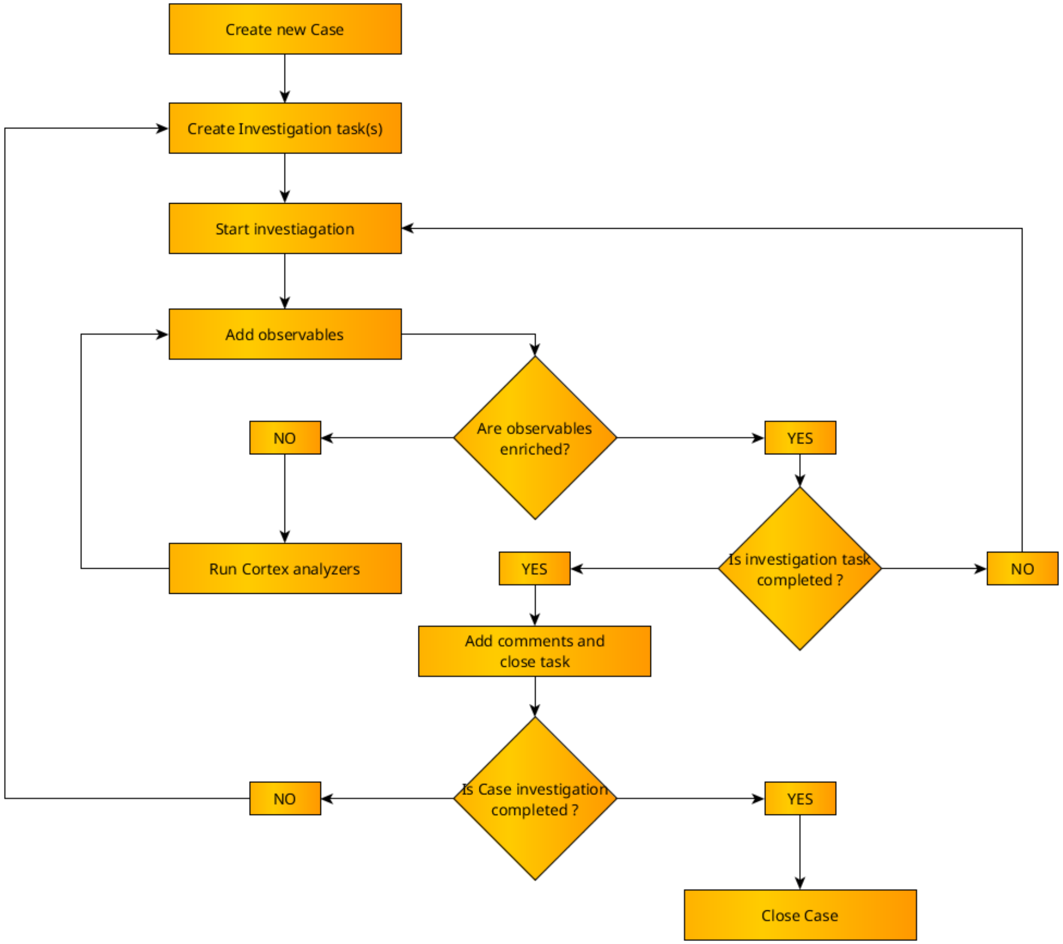
Cortex

- **Cortex - environment for applications called Analyzers**
- **Analyzers can be invoked from TheHive, directly from Cortex web interface, Cortex REST API or Cortex4py**
- **Analyzers output can be customized by templates**
- **Cortex engine has many built-in analyzers written in python**
- **Any programming language can be used for writing analyzer**
- **Easy to write own analyzers**
  - Definition: new_analyzer.json
  - Main script: new_analyzer.py
  - Optionally: requirements.txt

Short description of few built-in analyzers

- **CIRCLPassiveDNS: Check CIRCL's Passive DNS for a given domain.**
- **GoogleSafebrowsing: check URLs against Google Safebrowsing.**
- **MaxMind: geolocation.**
- **MISP Search: search for MISP events in one or several MISP instances containing the observable submitted as input.**
- **VirusTotal: look up files, URLs and hashes through VirusTotal.**
- **Yara: check files against YARA rules using yara-python.**

TheHive Workflow


Create new Case
Create Investigation task(s)
Start investiagation
Add observables
Are observables enriched?
NO
YES
Run Cortex analyzers
Is investigation task completed ?
YES
NO
Add comments and close task
Is Case investigation completed ?
NO
YES
Close Case

TheHive & Cortex Administration Module

Tasks

Basic TheHive Configuration

Let’s start with setting basic configuration options.

- **cd /opt/enisa/trainings-2019/admin/thehive**

To start the exercise type in

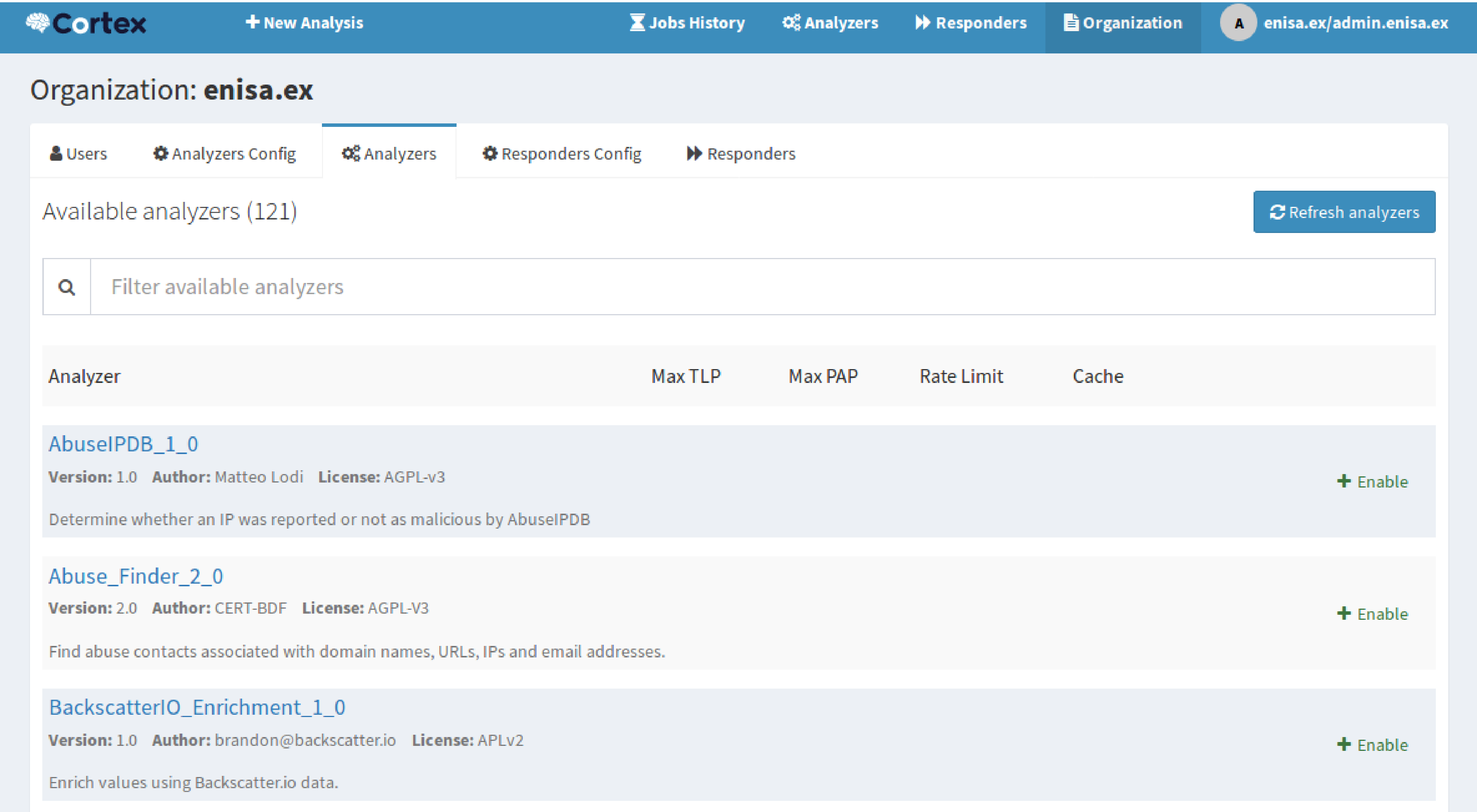
- **./start_exercise.sh**

Point your browser to thehive.enisa.ex and create admin account (admin:admin)

Basic Cortex Configuration

Point your browser to cortex.enisa.ex and create accounts and new organisation according to user manual.

Cortex analyzers Configuration

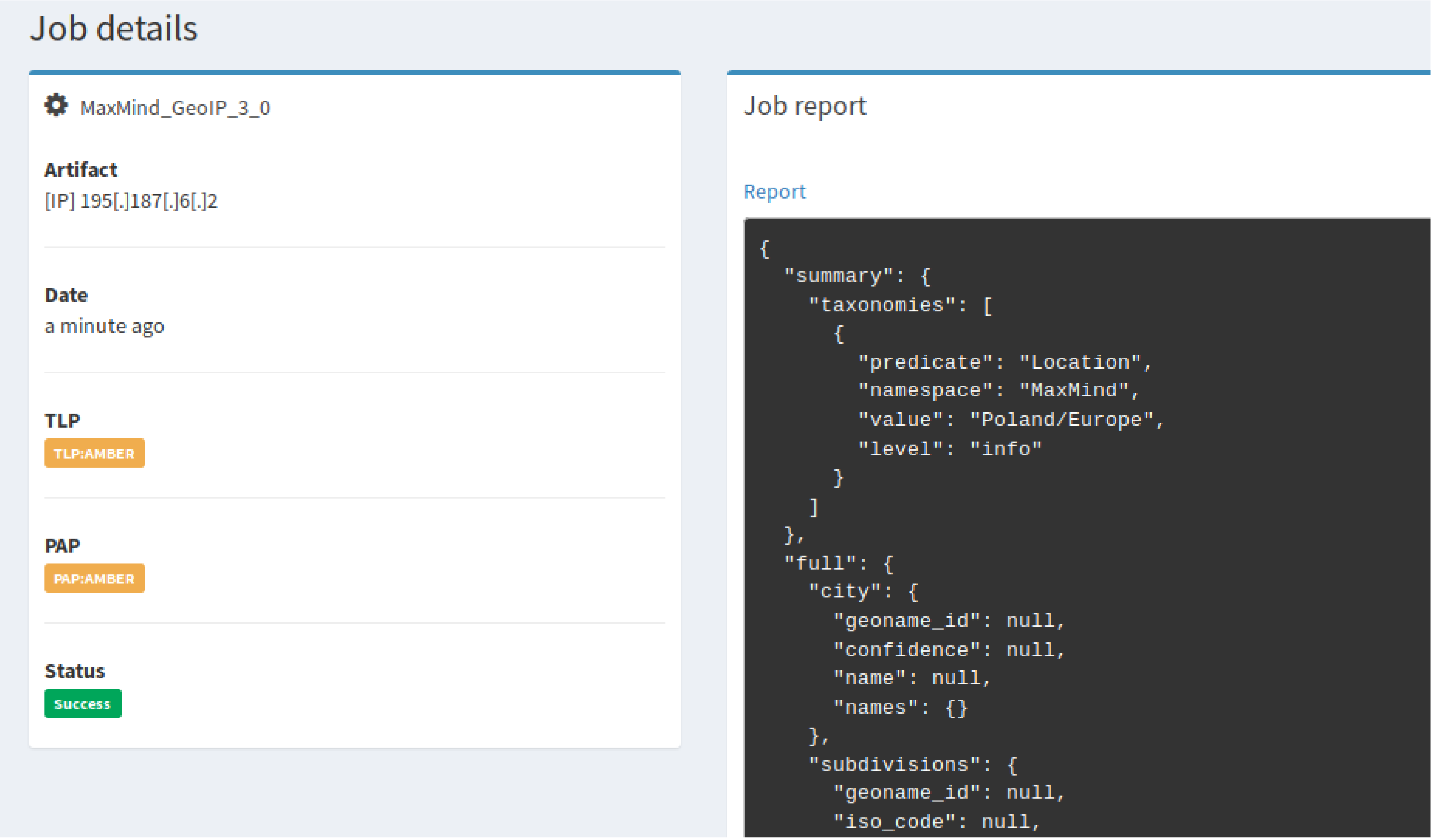


Organization: **enisa.ex**

Available analyzers (121)

Analyzer	Max TLP	Max PAP	Rate Limit	Cache	
AbuseIPDB_1_0 **Version:** 1.0 **Author:** Matteo Lodi **License:** AGPL-v3 Determine whether an IP was reported or not as malicious by AbuseIPDB					+ Enable
Abuse_Finder_2_0 **Version:** 2.0 **Author:** CERT-BDF **License:** AGPL-V3 Find abuse contacts associated with domain names, URLs, IPs and email addresses.					+ Enable
BackscatterIO_Enrichment_1_0 **Version:** 1.0 **Author:** brandon@backscatter.io **License:** APLv2 Enrich values using Backscatter.io data.					+ Enable

Cortex analyzers test



Job details

⚙ MaxMind_GeoIP_3_0

Artifact
[IP] 195[.]187[.]6[.]2

Date
a minute ago

TLP
TLP:AMBER

PAP
PAP:AMBER

Status
Success

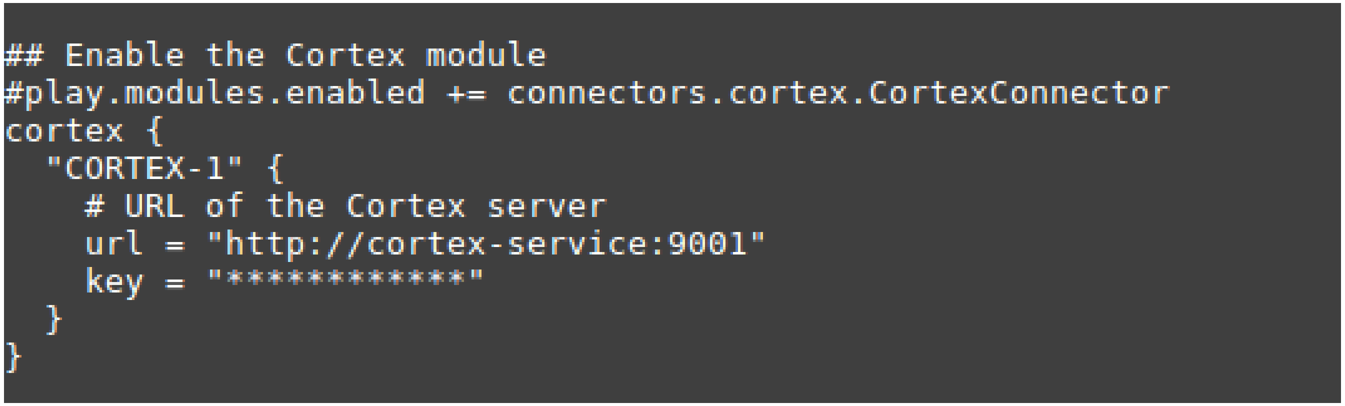
Job report

Report

```
{
  "summary": {
    "taxonomies": [
      {
        "predicate": "Location",
        "namespace": "MaxMind",
        "value": "Poland/Europe",
        "level": "info"
      }
    ]
  },
  "full": {
    "city": {
      "geoname_id": null,
      "confidence": null,
      "name": null,
      "names": {}
    },
    "subdivisions": {
      "geoname_id": null,
      "iso_code": null,
```

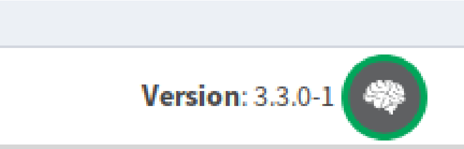
TheHive - Cortex integration

/opt/enisa/trainings-2019/admin/thehive/thehive-config/application.conf



```
## Enable the Cortex module
#play.modules.enabled += connectors.cortex.CortexConnector
cortex {
  "CORTEX-1" {
    # URL of the Cortex server
    url = "http://cortex-service:9001"
    key = "************"
  }
}
```

Remember to restart TheHive after saving file by executing ./restart_thehive.sh



Version: 3.3.0-1

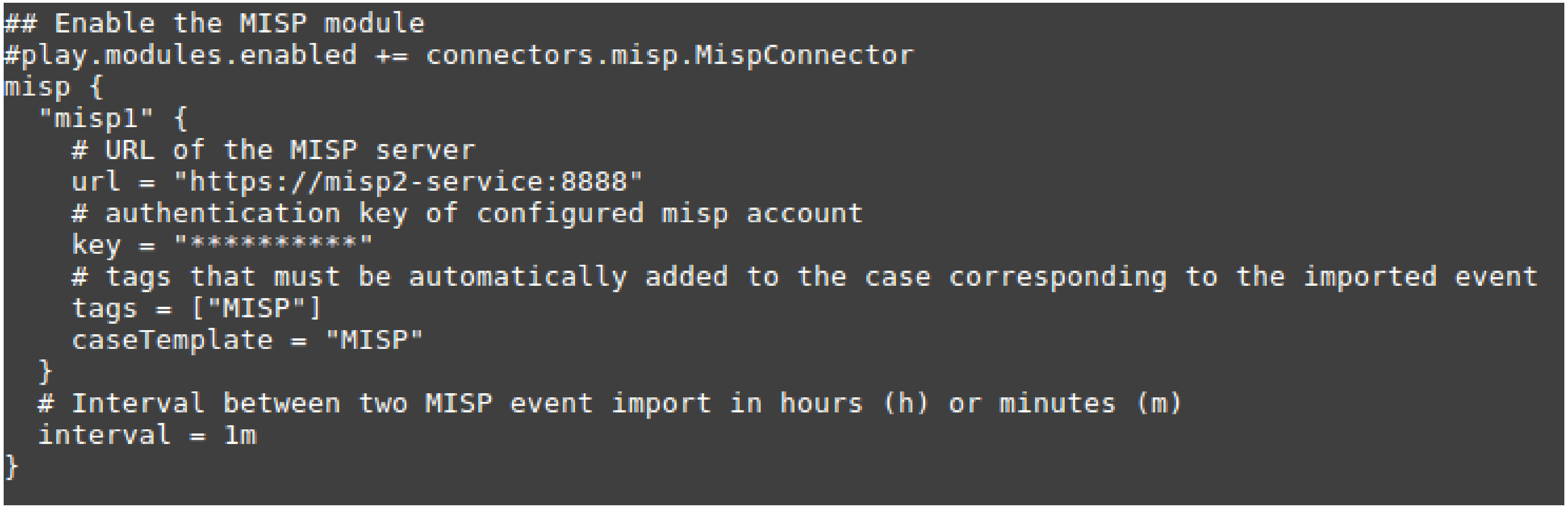
TheHive - Misp integration

Why integrate?

- **New events in MISP are automatically pulled by TheHive with given interval**
- **Taxonomy of MISP attributes are mapped into TheHive observables**
- **Easy to start new Case based on MISP event**
- **Search attributes in MISP instance directly from TheHive**
- **Export observables into several MISP instances**

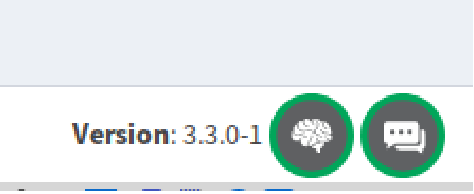
TheHive - Misp integration

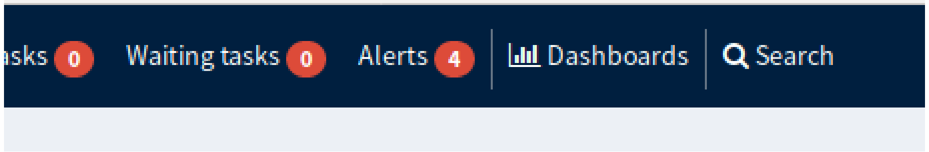
/opt/enisa/trainings-2019/admin/thehive/thehive-config/application.conf



```
## Enable the MISP module
#play.modules.enabled += connectors.misp.MispConnector
misp {
  "misp1" {
    # URL of the MISP server
    url = "https://misp2-service:8888"
    # authentication key of configured misp account
    key = "**********"
    # tags that must be automatically added to the case corresponding to the imported event
    tags = ["MISP"]
    caseTemplate = "MISP"
  }
  # Interval between two MISP event import in hours (h) or minutes (m)
  interval = 1m
}
```

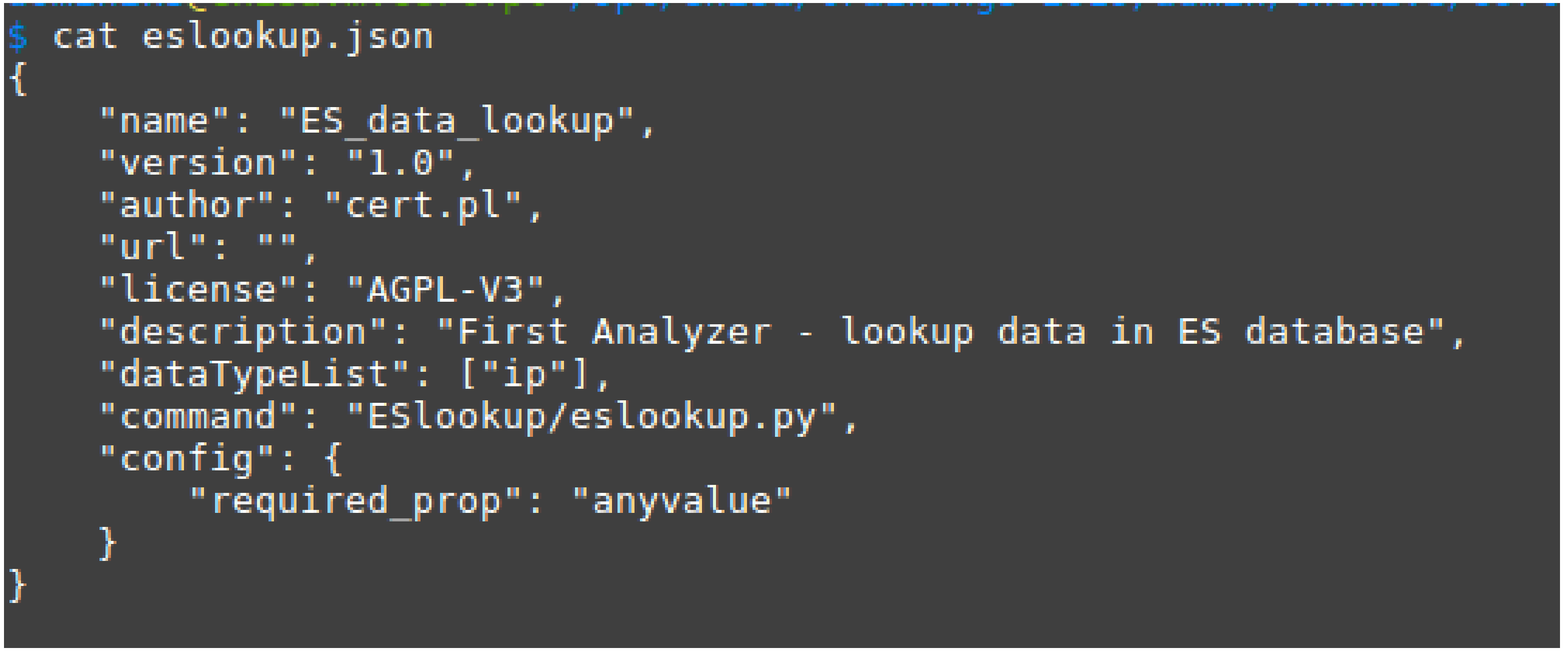
Remember to restart TheHive after saving file by executing ./restart_thehive.sh





Cortex custom analyzer - definition

/opt/enisa/trainings-2019/admin/thehive/cortex-analyzers/ESlookup/



```
$ cat eslookup.json
{
    "name": "ES_data_lookup",
    "version": "1.0",
    "author": "cert.pl",
    "url": "",
    "license": "AGPL-V3",
    "description": "First Analyzer - lookup data in ES database",
    "dataTypeList": ["ip"],
    "command": "ESlookup/eslookup.py",
    "config": {
        "required_prop": "anyvalue"
    }
}
```

Cortex custom analyzer - definition

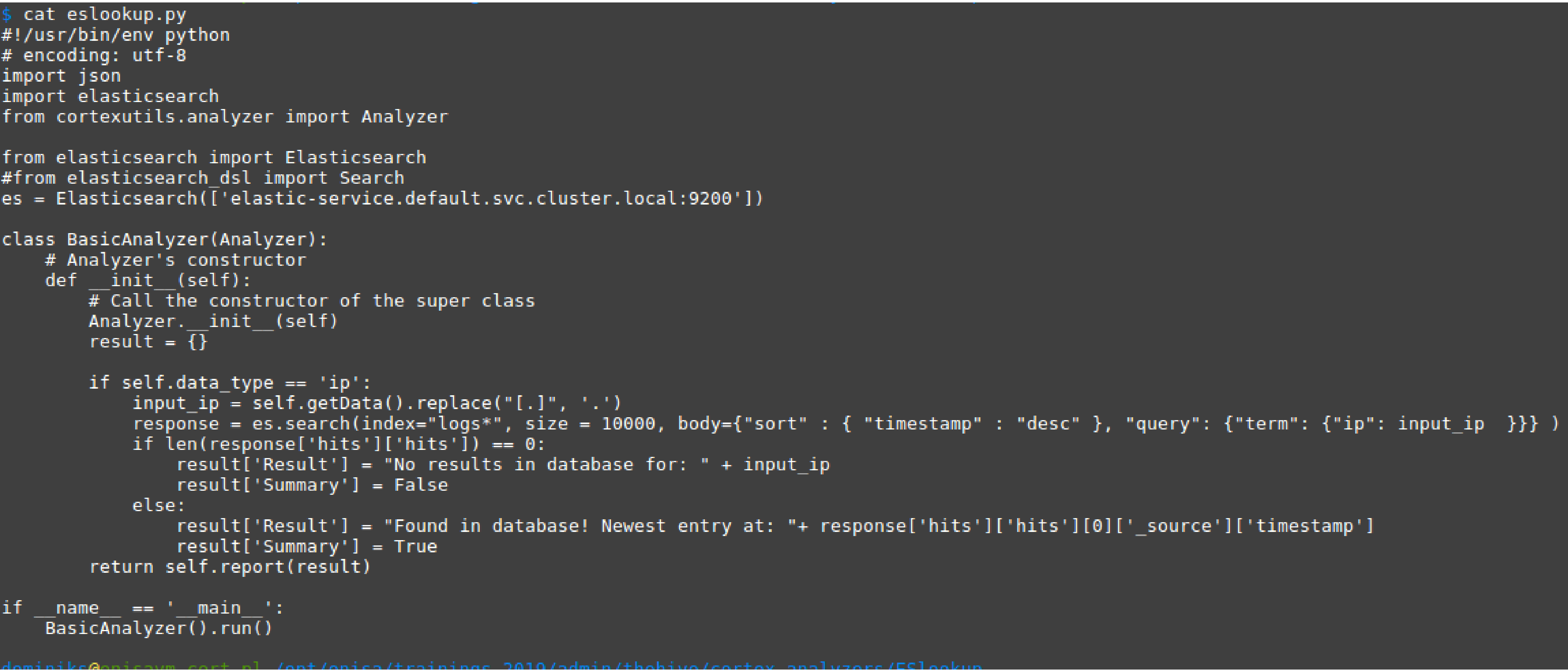
- **Each Analyzer has a config**
- **Analyzer definition file <name>.json may have section:**

```
"config": {
        "check_tlp": true,
        "max_tlp": 3
},
```

Why is it important?

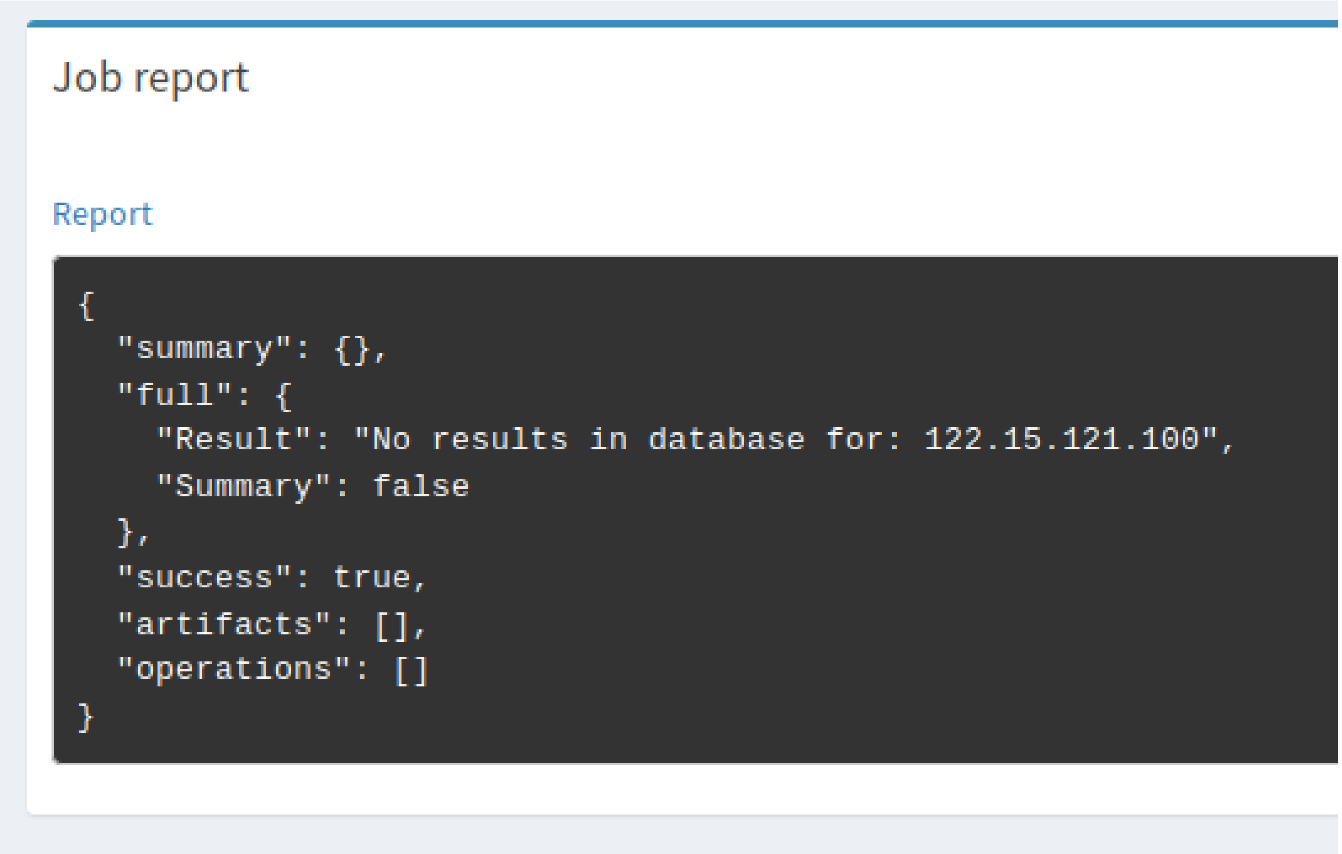
Cortex custom analyzer - implementation

/opt/enisa/trainings-2019/admin/thehive/cortex-analyzers/ESlookup/



```
$ cat eslookup.py
#!/usr/bin/env python
# encoding: utf-8
import json
import elasticsearch
from cortexutils.analyzer import Analyzer

from elasticsearch import Elasticsearch
#from elasticsearch_dsl import Search
es = Elasticsearch(['elastic-service.default.svc.cluster.local:9200'])

class BasicAnalyzer(Analyzer):
    # Analyzer's constructor
    def __init__(self):
        # Call the constructor of the super class
        Analyzer.__init__(self)
        result = {}

        if self.data_type == 'ip':
            input_ip = self.getData().replace("[.]", '.')
            response = es.search(index="logs*", size = 10000, body={"sort" : { "timestamp" : "desc" }, "query": {"term": {"ip": input_ip  }}} )
            if len(response['hits']['hits']) == 0:
                result['Result'] = "No results in database for: " + input_ip
                result['Summary'] = False
            else:
                result['Result'] = "Found in database! Newest entry at: "+ response['hits']['hits'][0]['_source']['timestamp']
                result['Summary'] = True
        return self.report(result)

if __name__ == '__main__':
    BasicAnalyzer().run()

dominiks@enisavm.cert.pl /opt/enisa/trainings-2019/admin/thehive/cortex-analyzers/ESlookup
```

Cortex custom analyzer

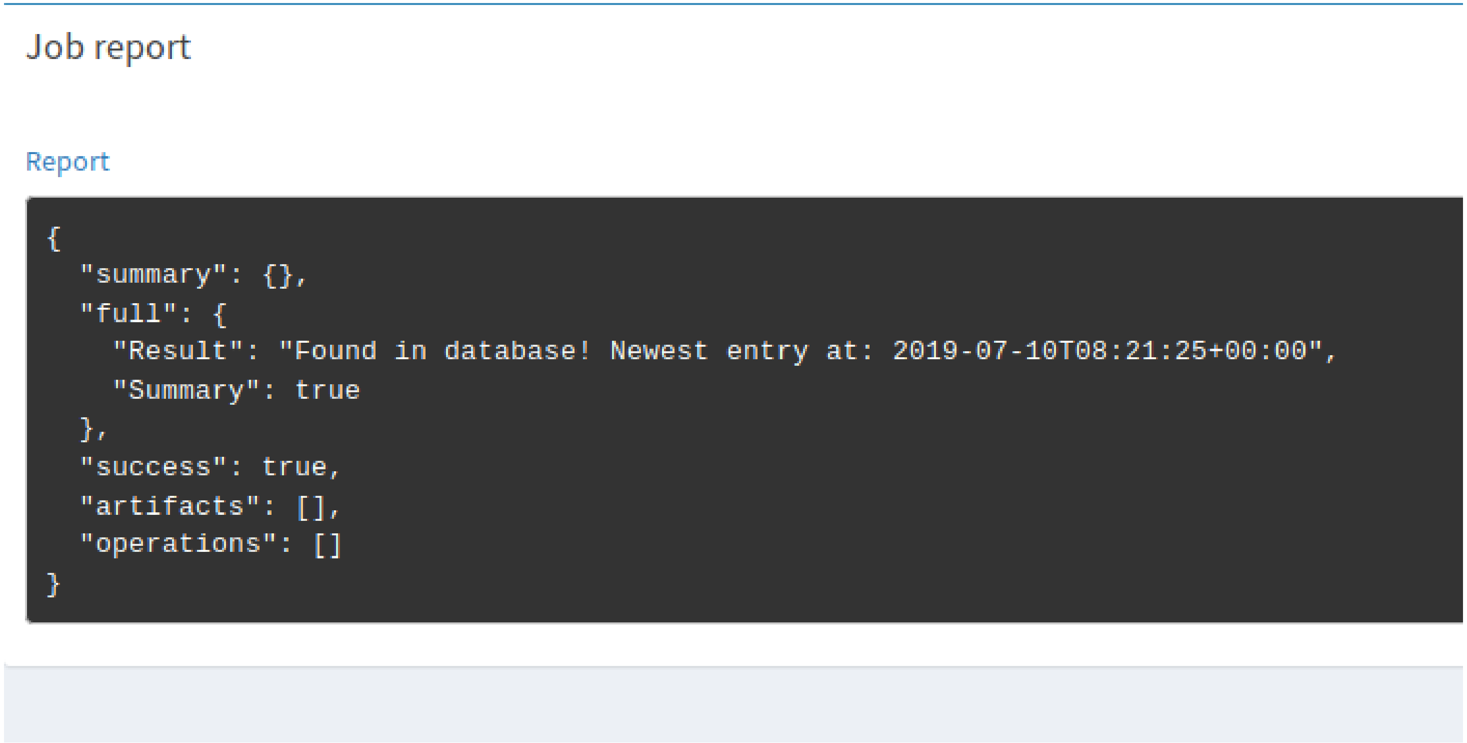


Job report

Report

```
{
  "summary": {},
  "full": {
    "Result": "No results in database for: 122.15.121.100",
    "Summary": false
  },
  "success": true,
  "artifacts": [],
  "operations": []
}
```

Cortex custom analyzer



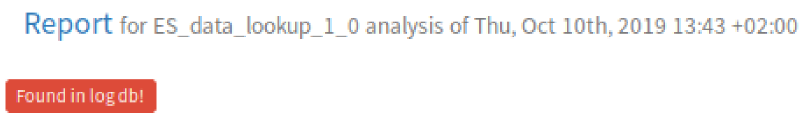
Job report

Report

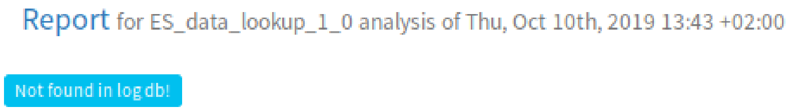
```
{
  "summary": {},
  "full": {
    "Result": "Found in database! Newest entry at: 2019-07-10T08:21:25+00:00",
    "Summary": true
  },
  "success": true,
  "artifacts": [],
  "operations": []
}
```

Report templates

- **Output from analyzers may be customized using report templates.**
- **They allow to show results using html/bootstrap instead of plain json.**



Report for ES_data_lookup_1_0 analysis of Thu, Oct 10th, 2019 13:43 +02:00



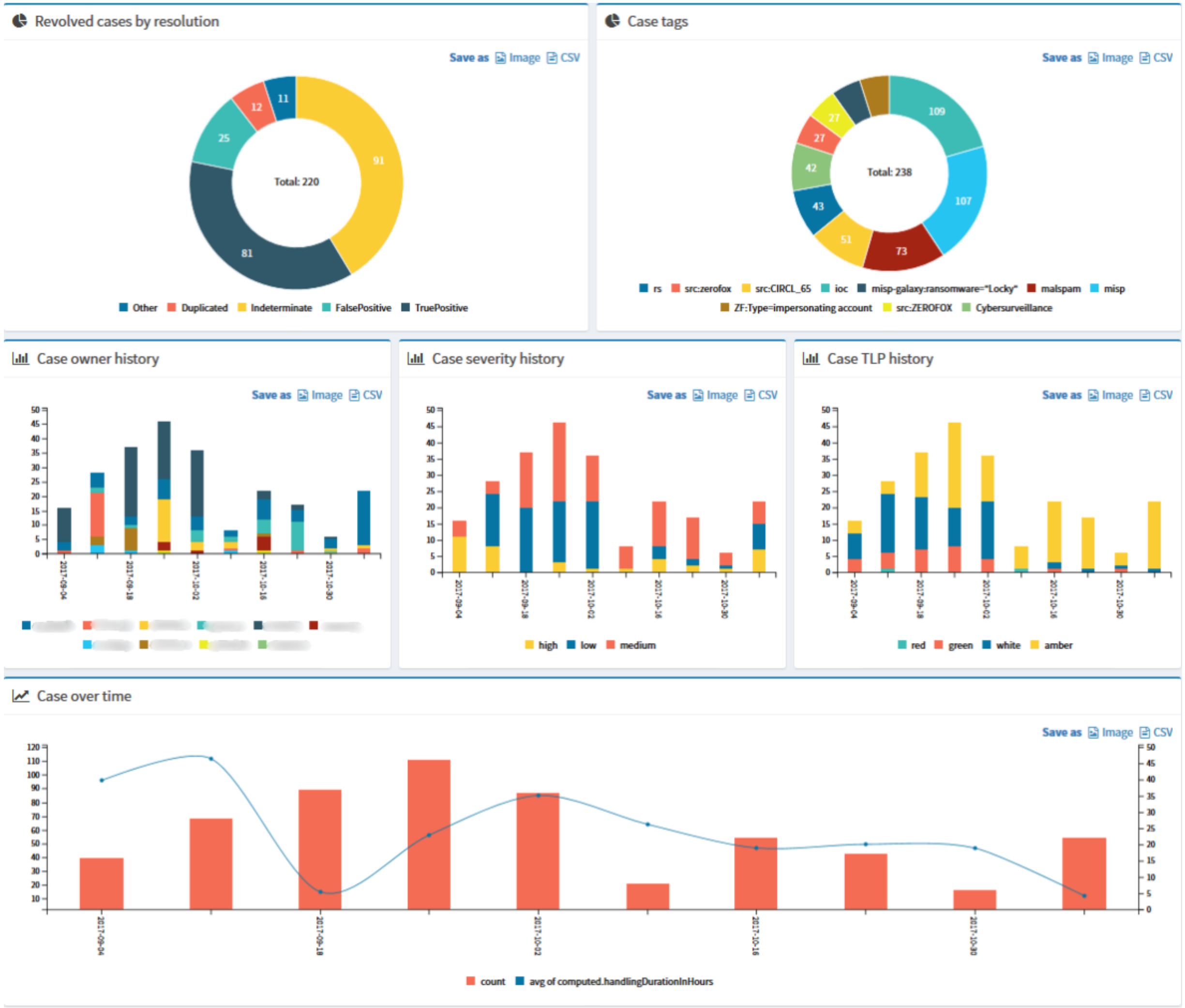
Report for ES_data_lookup_1_0 analysis of Thu, Oct 10th, 2019 13:43 +02:00

Case templates

Allow to create templated forms of Case creation.

- **Why it may be useful?**
- **Let’s create example case template**

Dashboards


Revolved cases by resolution
Save as Image CSV
11
12
25
91
Total: 220
81
Other
Duplicated
Indeterminate
FalsePositive
TruePositive
Case tags
Save as Image CSV
109
27
27
42
Total: 238
107
43
51
73
rs
src:zerofox
src:CIRCL_65
ioc
misp-galaxy:ransomware="Locky"
malspam
misp
ZF:Type=impersonating account
src:ZEROFOX
Cybersurveillance
Case owner history
Save as Image CSV
2017-09-04
2017-09-18
2017-10-02
2017-10-16
2017-10-30
Case severity history
Save as Image CSV
high
low
medium
Case TLP history
Save as Image CSV
red
green
white
amber
Case over time
Save as Image CSV
2017-09-04
2017-09-18
2017-10-02
2017-10-16
2017-10-30
count
avg of computed.handlingDurationInHours

CHAPTER 4

ORCHESTRATION OF CSIRT TOOLS

MISP ANALYST MODULE

MALWARE HUNTING & SHARING THREAT INTELLIGENCE

MISP Analyst Module

malware hunting & sharing threat intelligence

Introduction

Preconfigured states

For the exercise purposes, we prepared two states of the exercise that you can install by instructions provided in the next slides.

NOTE: More detailed instructions about all topics discussed in this presentation can be found in the student's handbook. Please open them now.

Misp bare

This state consists of two MISP systems.

One (*https://misp.enisa.ex*) is not configured at all.

This represents state after admin configuration.

- There are **taxonomies** and **galaxies** downloaded
- There are multiple events imported from open source of events
- One account is available with username: adm**in@admin.test** and password **FirstInstancePassword!**

Another instance (*https://misp2.enisa.ex*) contains data and minimal configuration.

Credentials: **admin@admin.test**:**SecondInstancePassword123!**

Misp configured

This represents both misps in condition after the exercise is finished.

Follow the steps in the student’s handbook to get to this stage from the misp-bare snapshot.

MISP Analyst Module

malware hunting & sharing threat intelligence

Exercise

Installation

Let’s start with setting basic configuration options.

- *cd /opt/enisa/trainings-2019/analyst/misp*
- To start the exercise type in *./start-exercise.sh*
- Navigate to your organization's MISP with web browser (*https://misp.enisa.ex*)
- You should be presented with preconfigured MISP instance that you can use as a playground in following exercises.

Events

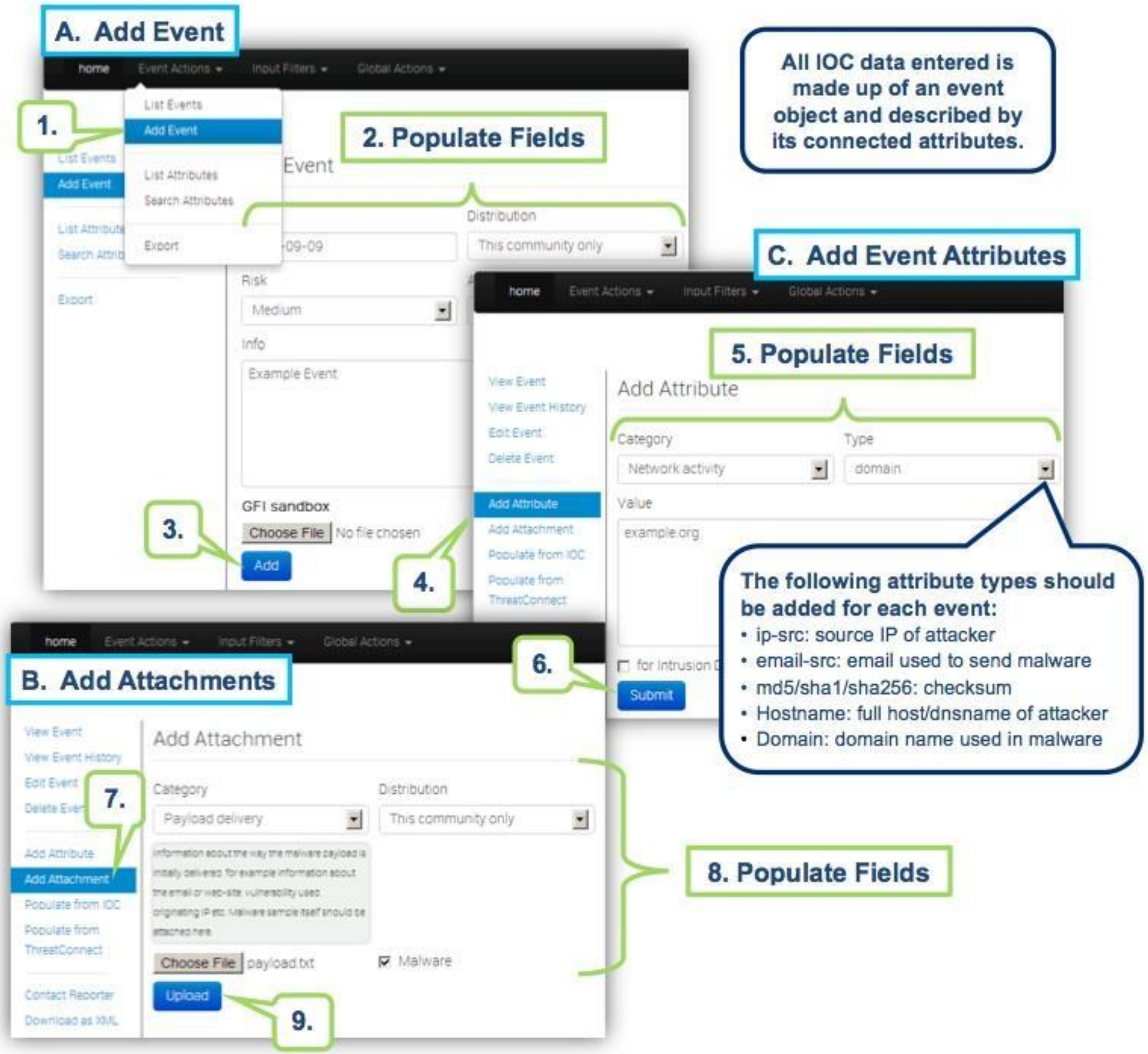
Events are the core of misp instance.

They allow you to manage, share and enrich intelligence of yours and others organisations.

The process of entering an event can be split into 3 phases, the **creation** of the event itself, **populating** it with attributes and attachments and finally **publishing** it.

- Add an event in **Event Actions** -> **Add Event**

Events



A. Add Event

2. Populate Fields

Event

Date: …-09-09

Distribution: This community only

Risk: Medium

Info: Example Event

GFI sandbox: No file chosen

C. Add Event Attributes

5. Populate Fields

Add Attribute

Category: Network activity

Type: domain

Value: example.org

☐ for Intrusion D…

B. Add Attachments

Add Attachment

Category: Payload delivery

Distribution: This community only

information about the way the malware payload is initially delivered, for example information about the email or web-site, vulnerability used, originating IP etc. Malware sample itself should be attached here.

payload.txt

☑ Malware

8. Populate Fields

All IOC data entered is made up of an event object and described by its connected attributes.

The following attribute types should be added for each event:

- ip-src: source IP of attacker
- email-src: email used to send malware
- md5/sha1/sha256: checksum
- Hostname: full host/dnsname of attacker
- Domain: domain name used in malware

Events

Ex. 1: Adding an event.

To add an event, click the Add Event option when on the List Events view. **Event Actions** -> **Add Event**

- **Distribution**: Defines how far in the chain of synchronized misps the event is gonna be published.
- **Analysis**: Defines if the event is in ongoing analysis or it's analysis has already completed.
- **Event info**: Description of the event, concise info of what happened, what the event is about.
- **Extends event**: MISP allows for correlation of events, in this field you can put UUIDs of other events that correlate to this incident.

Events

Ex. 1: Adding an event.

Event can store a lot of informations, those include **tags**, **attributes**, **related events**, **correlations** and so on.

Attributes are very important part of an event, they contain informations such as *IoCs*, *C&C* addresses, *md5* hashes, or other additional information.
There are multiple types of attributes.

Follow the student’s handbook to some of those attributes correctly.

Events

Ex. 2 - Search and correlation

Try to find all unclassified events in MISP that may be correlated in any way with the event you added in the previous exercise.

- Search by **file hash** or **IP**
- Try **View Correlation Graph**
- **Related Events** is a shortcut for searching by attributes

Galaxies

In MISP, galaxies are used to express a large object called cluster.

They are formed by elements (key:value pairs). Default vocabularies are available in MISP galaxy – they can be overwritten, replaced or updated.

- To add galaxy to the event go to the detailed event view
- Check what you can do with galaxies on your event
- To add galaxy to the event go to event view and click **Galaxies** -> **Add**

Taxonomies

Taxonomy is a group of „machine tags” used to tag events and attributes.

Every tag is composed of a namespace (mandatory), a predicate (mandatory) and a value (optional).

Example: osint:source-type=”blog-post” (osint - namespace, source-type - predicate, “blog-post” - value).

- These machine tags are often called **triple tag** due to their format.
- Add taxonomy to your event

CHAPTER 5

ORCHESTRATION OF CSIRT TOOLS

INTELMQ ANALYST MODULE

LOG ANALYSIS

IntelMQ Analyst Module

log analysis

Introduction

Introduction

This is an independent scenario focused on analysis, correlating and monitoring of logs collected through various systems and sources

IntelMQ

IntelMQ is a message queue for CERTS, SOCs and other security teams designed for collecting and processing security feeds. It is a community project, designed and used mostly by European CERTs/CSIRTs.

IntelMQ – entities

- Collectors - produce messages and pass them further into the system.
- Parsers - convert unstructured data into structured messages
- Experts - operate on parsed data and enrich or change it
- Outputs - send parsed data to other systems.

IntelMQ – nodes we’ll use in excercise

- File Collector: collector, that cyclically reads data from a file on the disk and passes them into the system
- JSON-Parser: parser, that reads JSON-serialised messages from input and converts them into a structured format understood by the IntelMQ
- Abusech-IP-Parser: another parser, but instead of JSON messages it was created for a specific feed - AbusechIP.
- Deduplicator-Expert: keeps events in a temporary database for configurable amount of time and drops already seen ones.
- Elasticsearch-Output: quite straightforward - stores processed events in a configured elasticsearch database.

Ensure that DNS is configured properly

Ensure that DNS is configured properly, and subdomains of .enisa.ex exist:

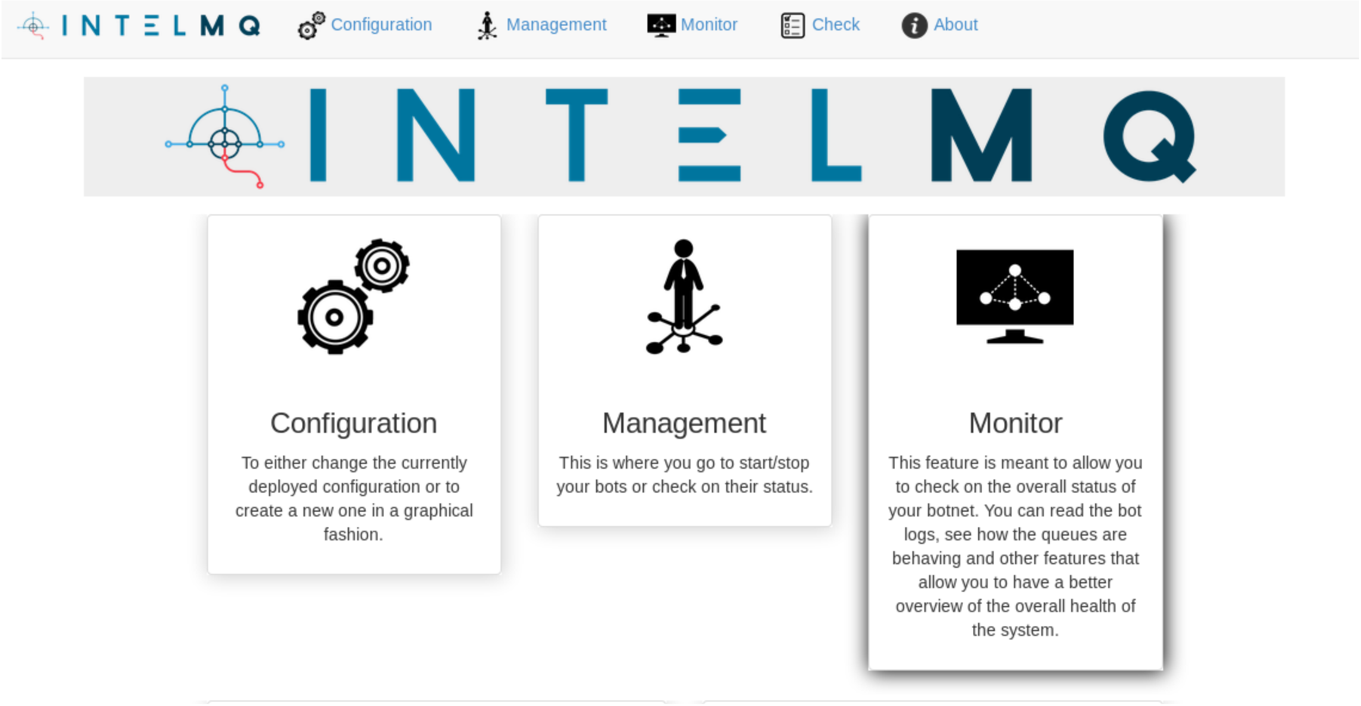
```
$ dig -ta +short intelmq.enisa.ex

127.0.0.1  # or any other valid IPv4

$ dig -ta +short kibana.enisa.ex

127.0.0.1  # or any other valid IPv4
```

Start the exercise

```
cd /opt/enisa/trainings-2019/analyst/intelmq/

$ helm install intelmq/
```

Ensure that elasticsearch works correctly

Point your browser to http://intelmq.enisa.ex. You should see the following:



Configuration

To either change the currently deployed configuration or to create a new one in a graphical fashion.

Management

This is where you go to start/stop your bots or check on their status.

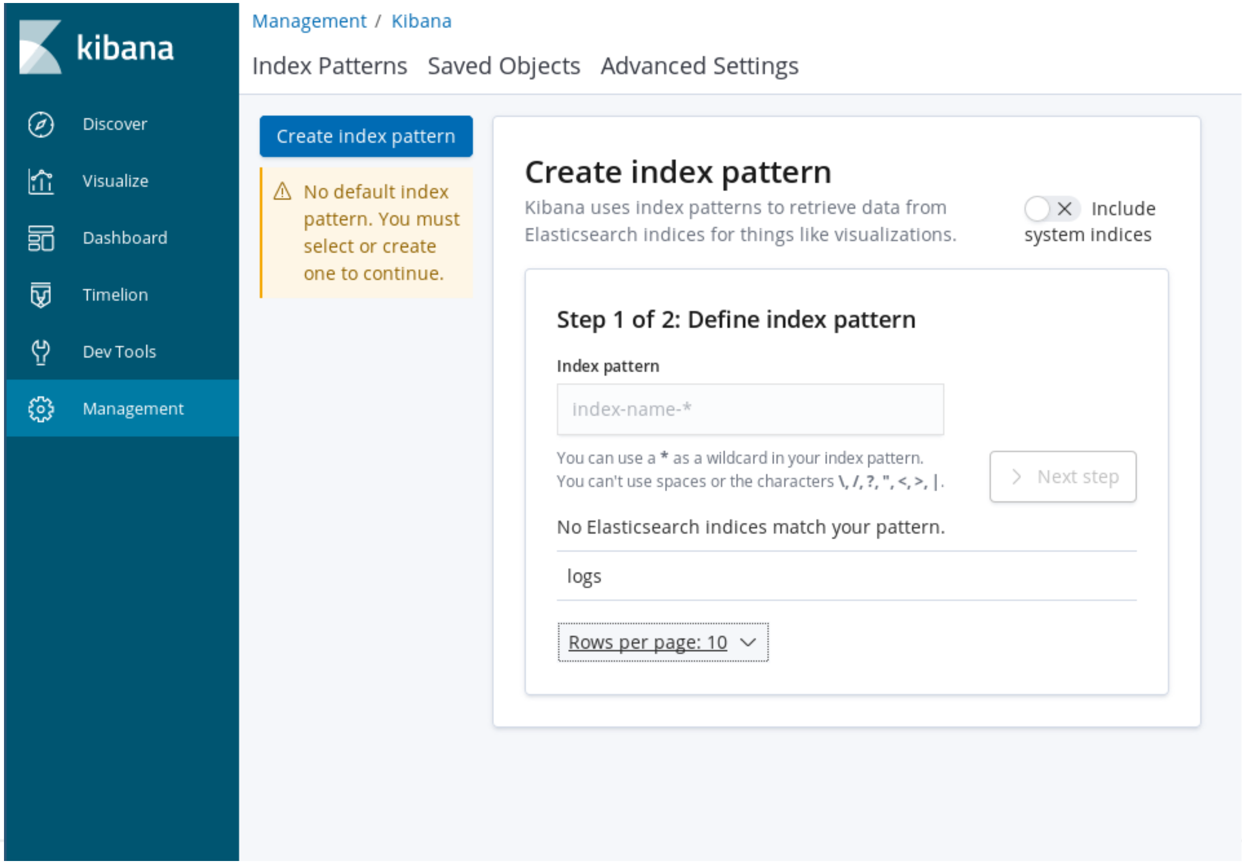
Monitor

This feature is meant to allow you to check on the overall status of your botnet. You can read the bot logs, see how the queues are behaving and other features that allow you to have a better overview of the overall health of the system.

If you see nginx 503 error instead, you have to wait a bit longer.

Ensure that kibana works correctly

Point your browser to http://kibana.enisa.ex



⚠ No default index pattern. You must select or create one to continue.

Create index pattern

Kibana uses index patterns to retrieve data from Elasticsearch indices for things like visualizations.

✕ Include system indices

Step 1 of 2: Define index pattern

Index pattern

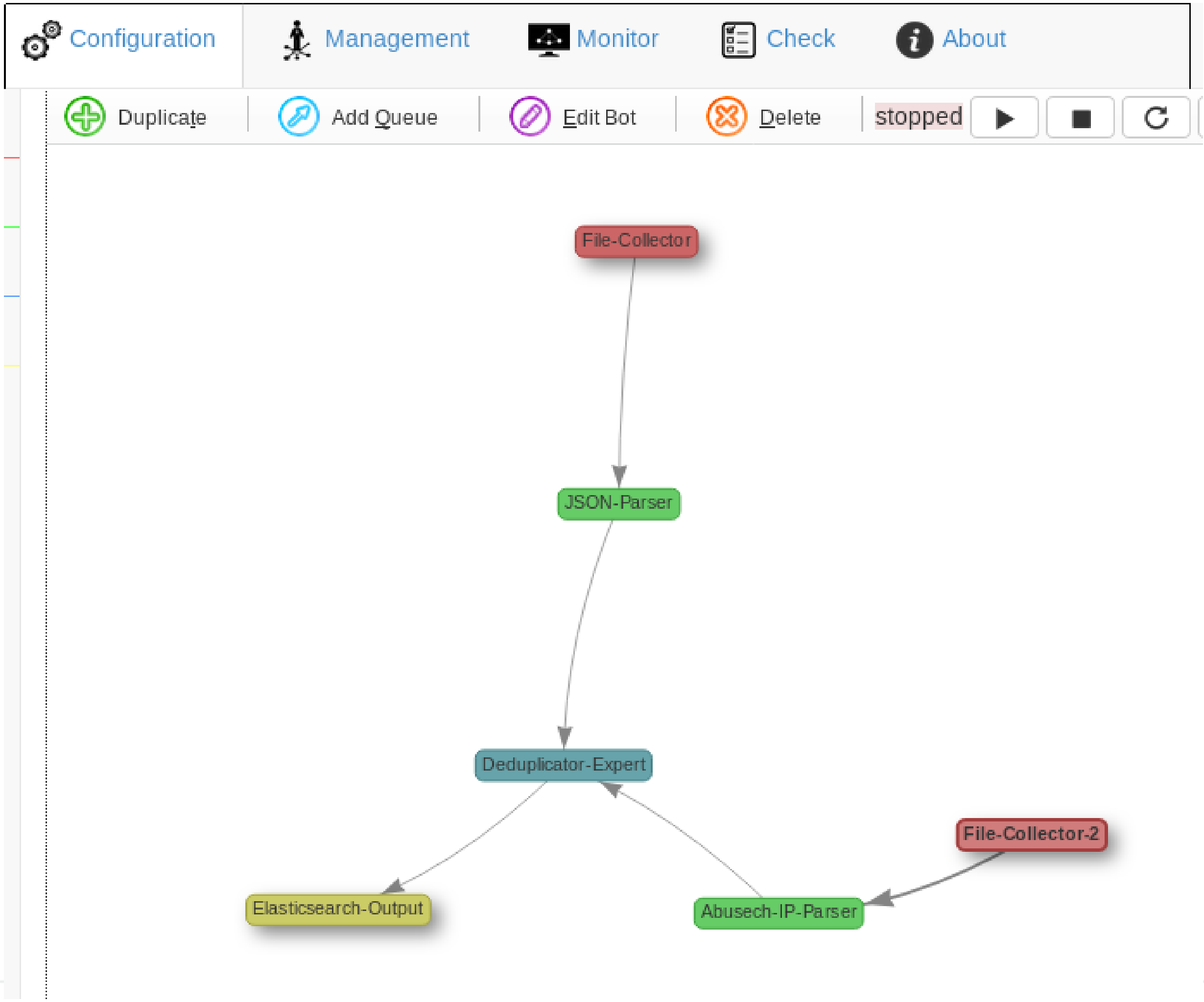
index-name-*

You can use a * as a wildcard in your index pattern.
You can't use spaces or the characters \, /, ?, ", <, >, |.

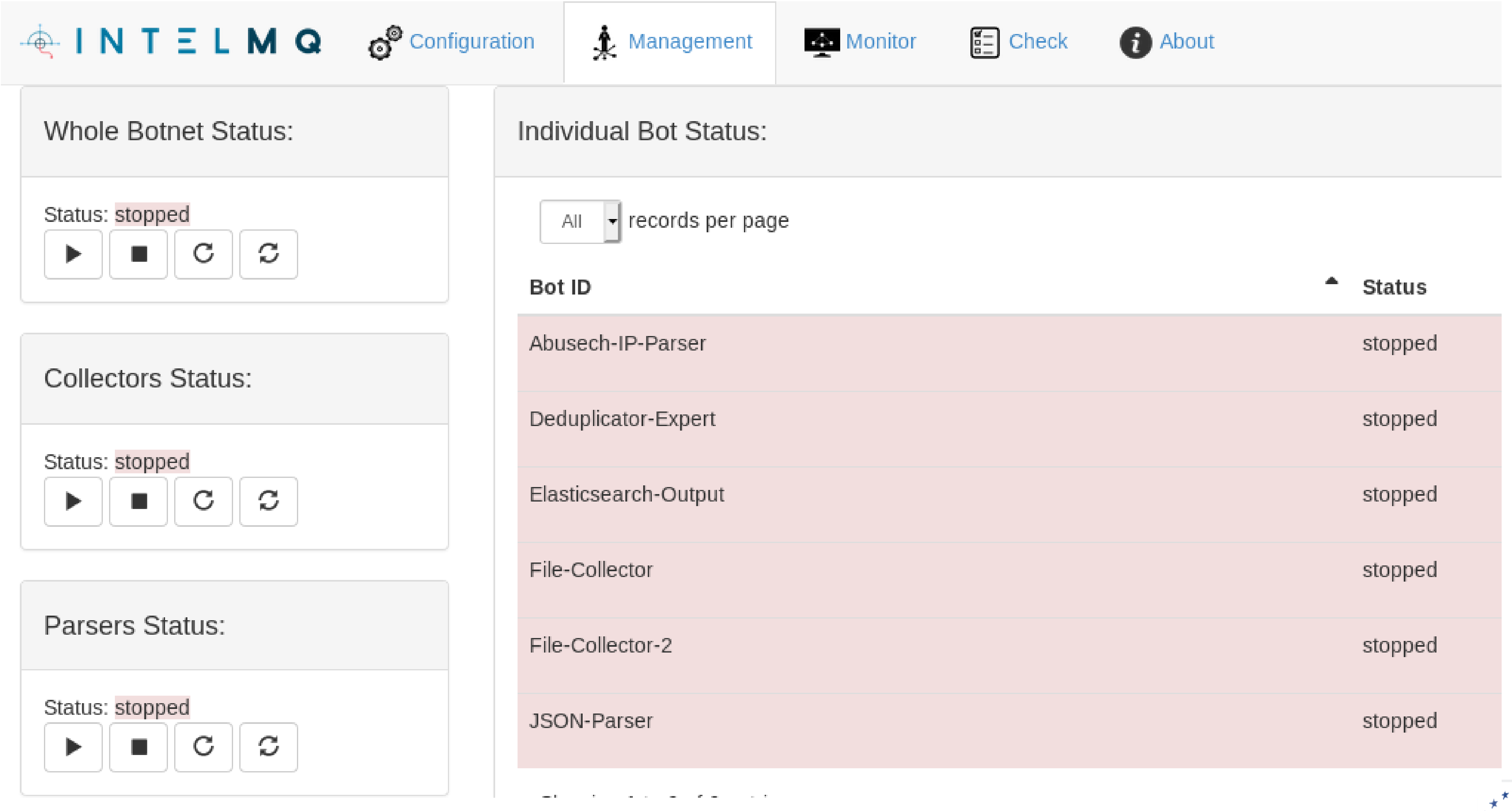
No Elasticsearch indices match your pattern.

logs

Get familiar with IntelMQ pipeline



Start the botnet



Whole Botnet Status:

Status: stopped

Collectors Status:

Status: stopped

Parsers Status:

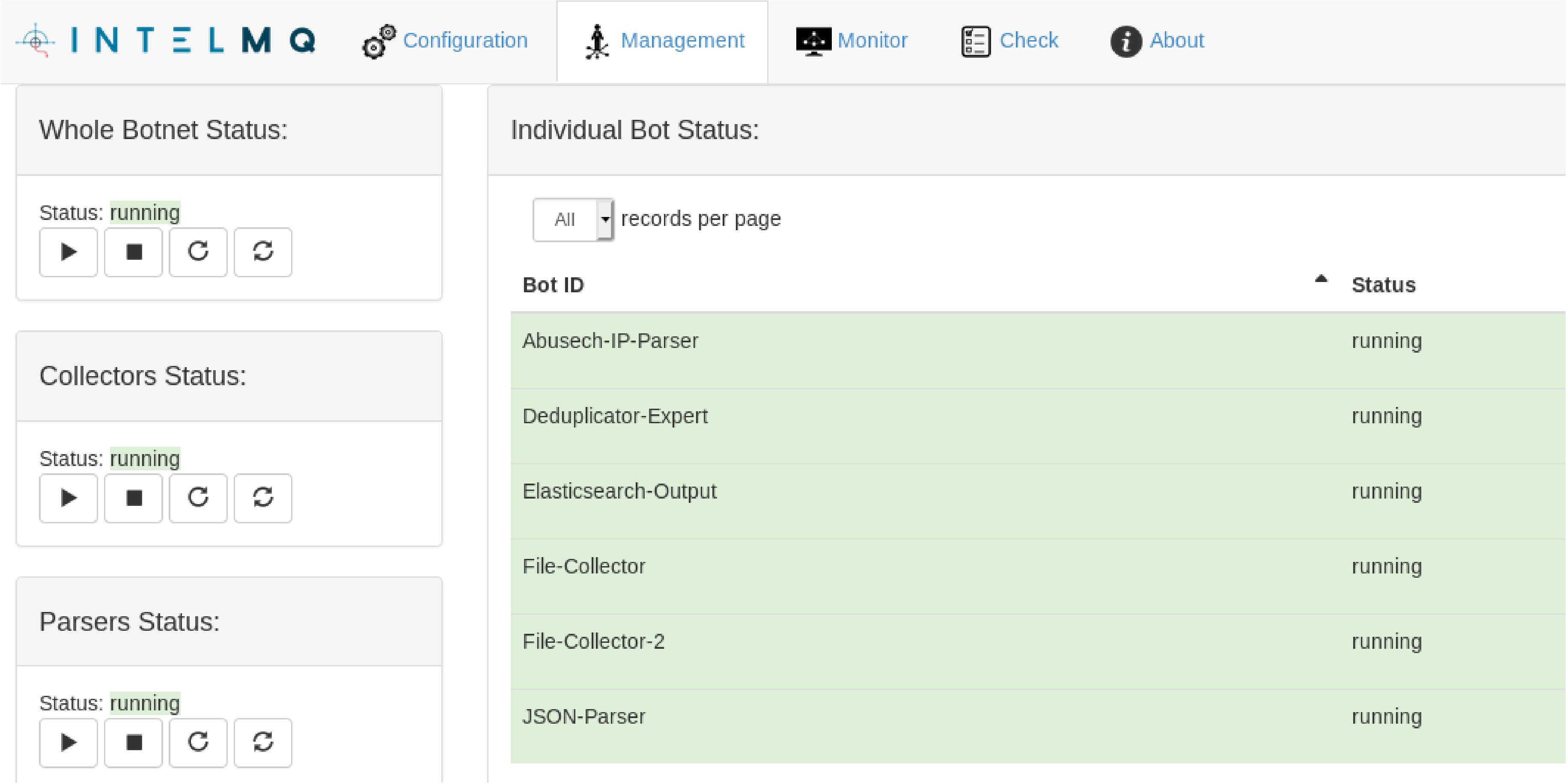
Status: stopped

Individual Bot Status:

All records per page

Bot ID	Status
Abusech-IP-Parser	stopped
Deduplicator-Expert	stopped
Elasticsearch-Output	stopped
File-Collector	stopped
File-Collector-2	stopped
JSON-Parser	stopped

Start the botnet



Whole Botnet Status:

Status: running

Collectors Status:

Status: running

Parsers Status:

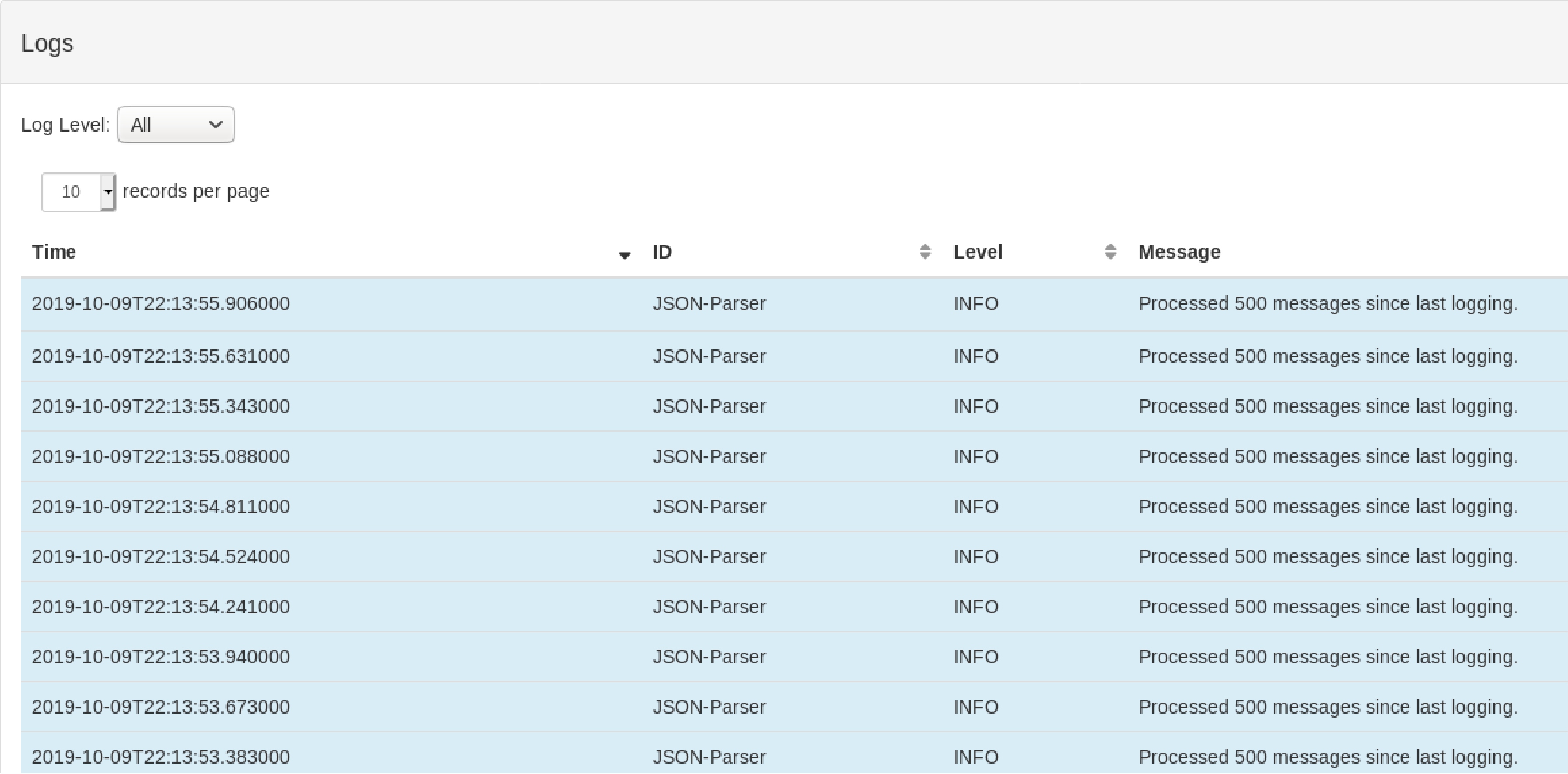
Status: running

Individual Bot Status:

All records per page

Bot ID	Status
Abusech-IP-Parser	running
Deduplicator-Expert	running
Elasticsearch-Output	running
File-Collector	running
File-Collector-2	running
JSON-Parser	running

Start the botnet



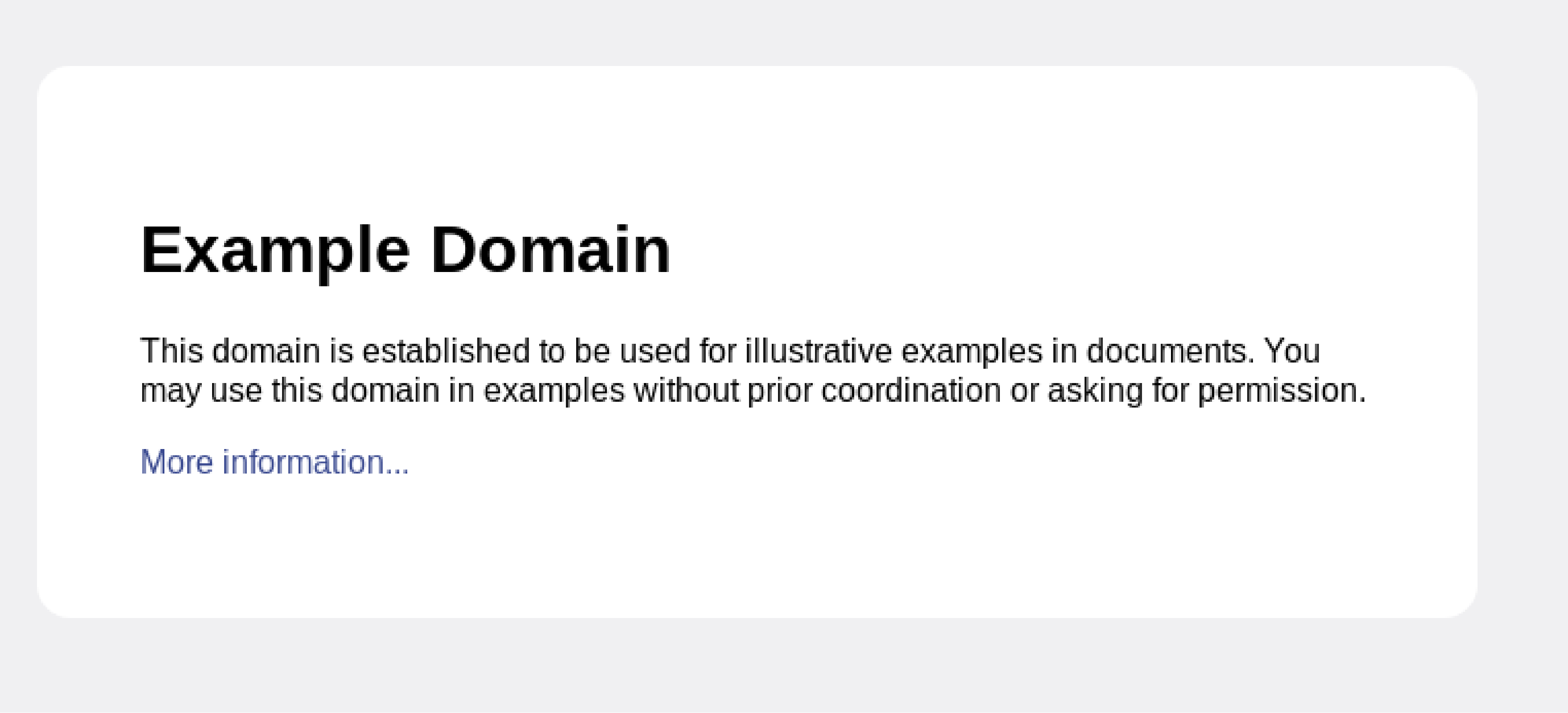
Logs

Log Level: All

10 records per page

Time	ID	Level	Message
2019-10-09T22:13:55.906000	JSON-Parser	INFO	Processed 500 messages since last logging.
2019-10-09T22:13:55.631000	JSON-Parser	INFO	Processed 500 messages since last logging.
2019-10-09T22:13:55.343000	JSON-Parser	INFO	Processed 500 messages since last logging.
2019-10-09T22:13:55.088000	JSON-Parser	INFO	Processed 500 messages since last logging.
2019-10-09T22:13:54.811000	JSON-Parser	INFO	Processed 500 messages since last logging.
2019-10-09T22:13:54.524000	JSON-Parser	INFO	Processed 500 messages since last logging.
2019-10-09T22:13:54.241000	JSON-Parser	INFO	Processed 500 messages since last logging.
2019-10-09T22:13:53.940000	JSON-Parser	INFO	Processed 500 messages since last logging.
2019-10-09T22:13:53.673000	JSON-Parser	INFO	Processed 500 messages since last logging.
2019-10-09T22:13:53.383000	JSON-Parser	INFO	Processed 500 messages since last logging.

Honeypot



Example Domain

This domain is established to be used for illustrative examples in documents. You may use this domain in examples without prior coordination or asking for permission.

More information...

Honeypot logs

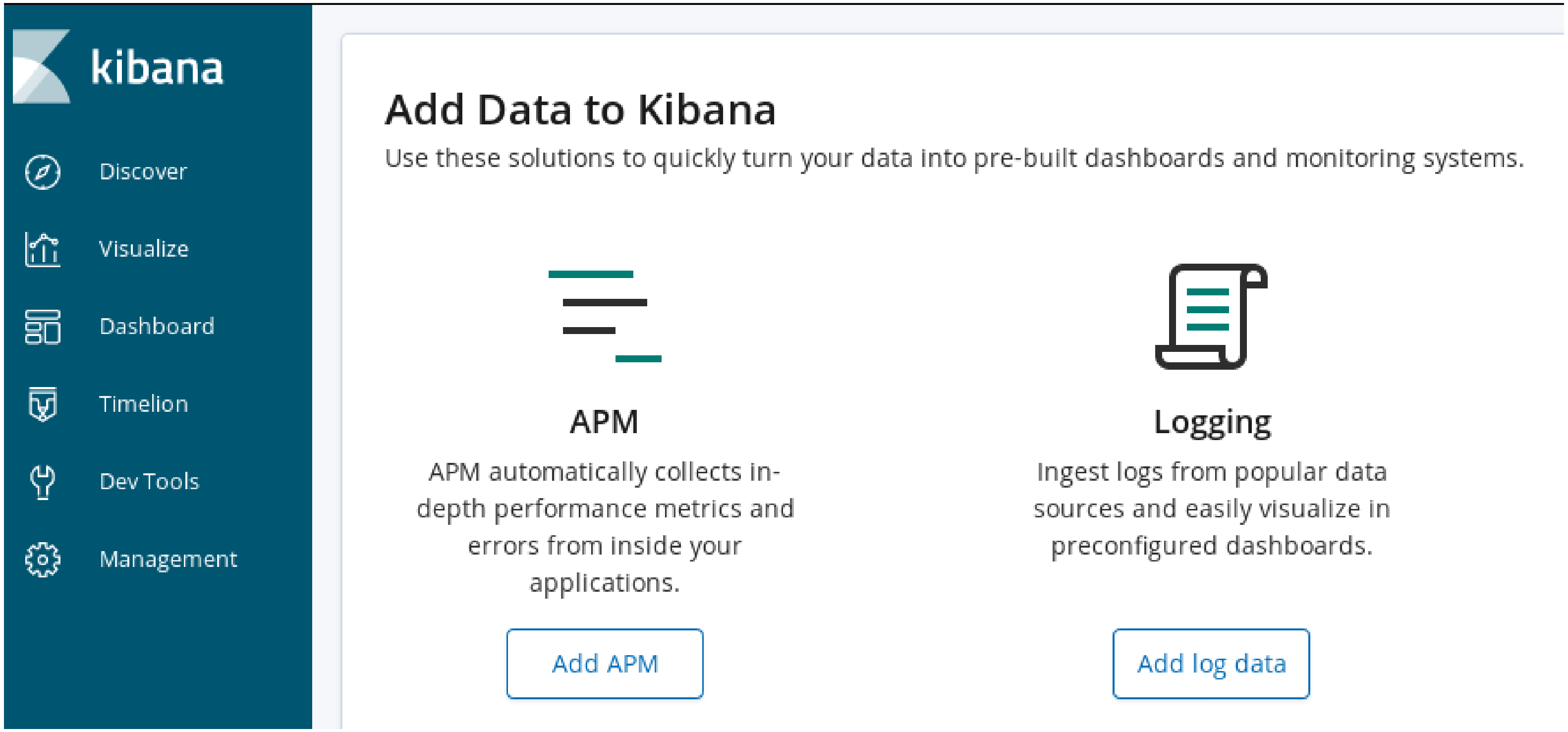
```
2019-09-10 16:14:45 INFO:snare.server:handle_request: Request path: /

2019-09-10 16:14:45 INFO:aiohttp.access:log: 10.1.1.1 [10/Sep/2019:16:14:45 +0000] "GET /
HTTP/1.1" 200 1422 "-" "Mozilla/5.0 (Windows NT 10.0; rv:68.0) Gecko/20100101 Firefox/68.0"

2019-09-10 16:14:45 INFO:snare.server:handle_request: Request path: /

2019-09-10 16:14:45 INFO:aiohttp.access:log: 10.1.1.1 [10/Sep/2019:16:14:45 +0000] "GET /
HTTP/1.1" 200 1362 "-" "Mozilla/5.0 (Windows NT 10.0; rv:68.0) Gecko/20100101 Firefox/68.0"

2019-09-10 16:14:46 INFO:snare.server:handle_request: Request path: /

2019-09-10 16:14:46 INFO:aiohttp.access:log: 10.1.1.1 [10/Sep/2019:16:14:46 +0000] "GET /
HTTP/1.1" 200 1362 "-" "Mozilla/5.0 (Windows NT 10.0; rv:68.0) Gecko/20100101 Firefox/68.0"

2019-09-10 16:14:51 INFO:snare.server:handle_request: Request path: /

2019-09-10 16:14:51 INFO:aiohttp.access:log: 10.1.1.1 [10/Sep/2019:16:14:51 +0000] "GET /
HTTP/1.1" 200 1362 "-" "Mozilla/5.0 (Windows NT 10.0; rv:68.0) Gecko/20100101 Firefox/68.0"
```

Logs converter

```
/opt/enisa/trainings-2019/analyst/intelmq/shared
$ python3 parse_logs.py snare.log snare_log.json
```

Kibana

open http://kibana.enisa.ex in your browser



Add Data to Kibana

Use these solutions to quickly turn your data into pre-built dashboards and monitoring systems.

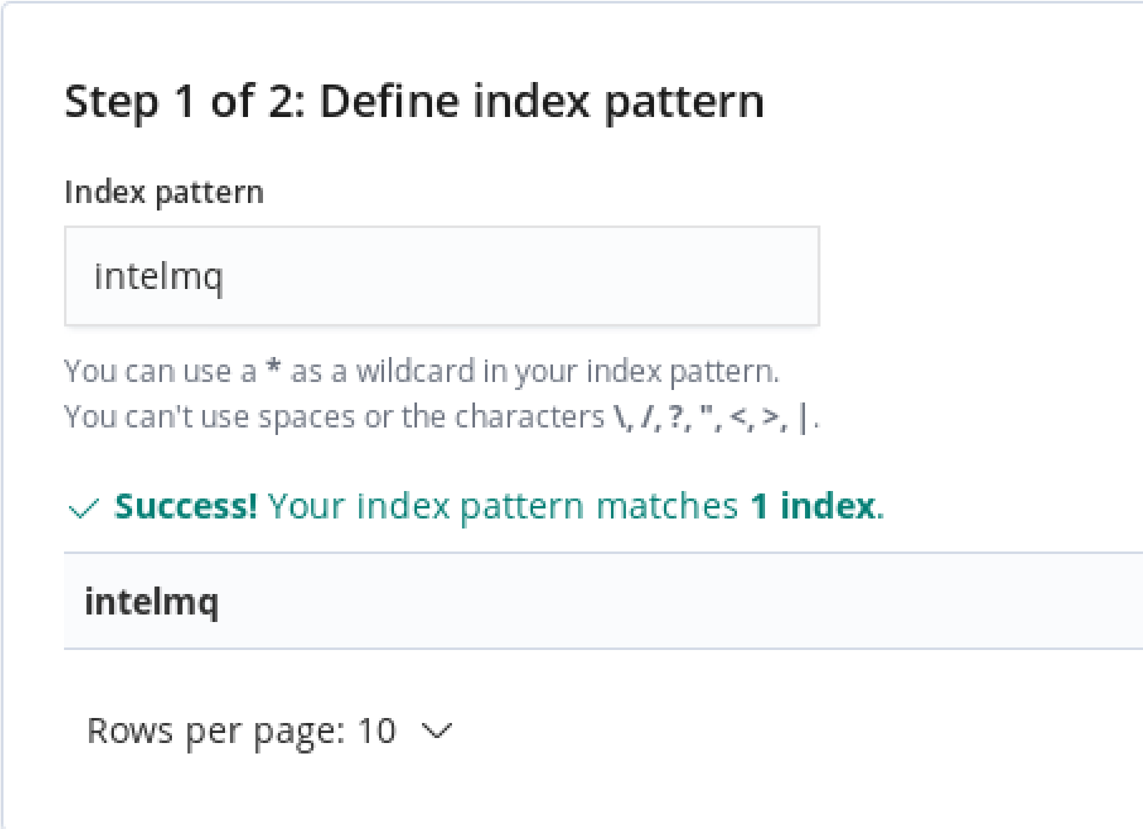
APM

APM automatically collects in-depth performance metrics and errors from inside your applications.

Logging

Ingest logs from popular data sources and easily visualize in preconfigured dashboards.

Kibana



Step 1 of 2: Define index pattern

Index pattern

intelmq

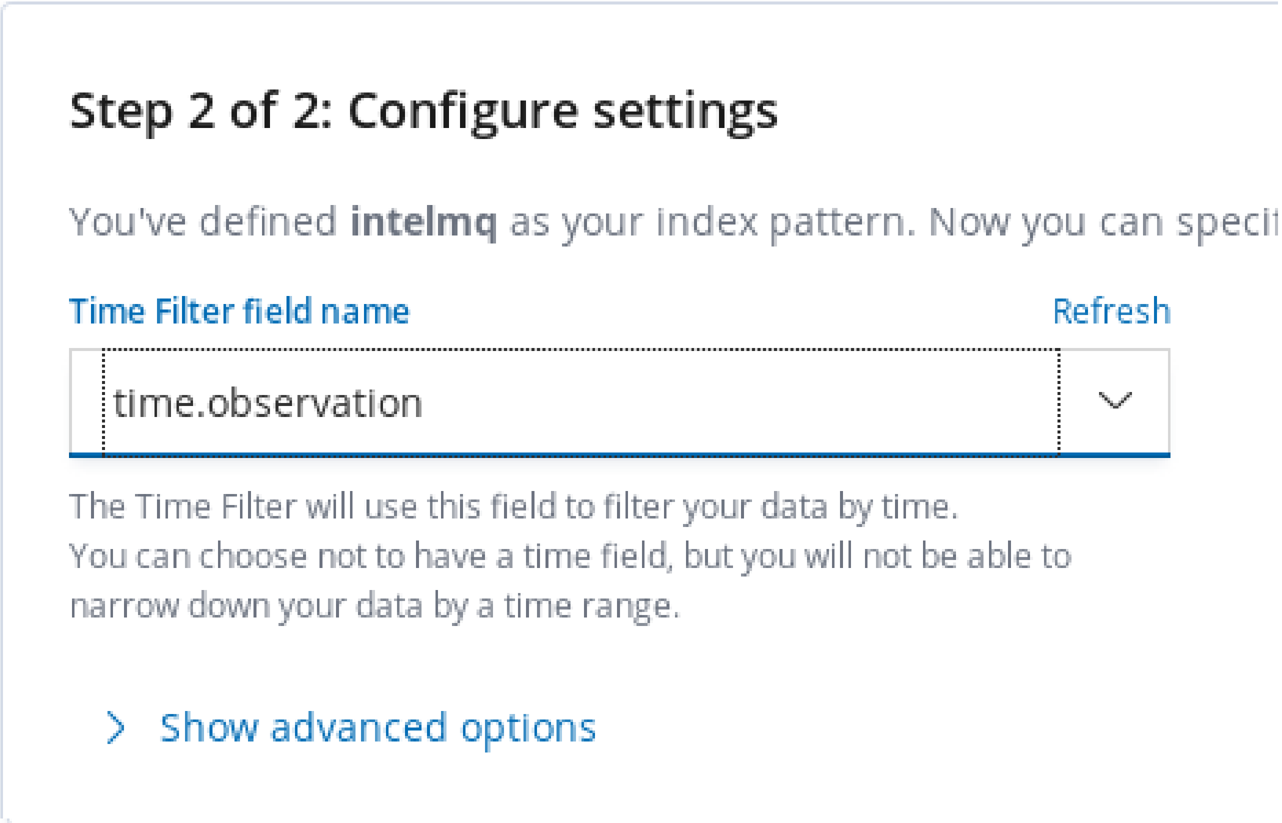
You can use a * as a wildcard in your index pattern.
You can't use spaces or the characters \, /, ?, ", <, >, | .

✓ **Success!** Your index pattern matches **1 index**.

intelmq

Rows per page: 10

Kibana



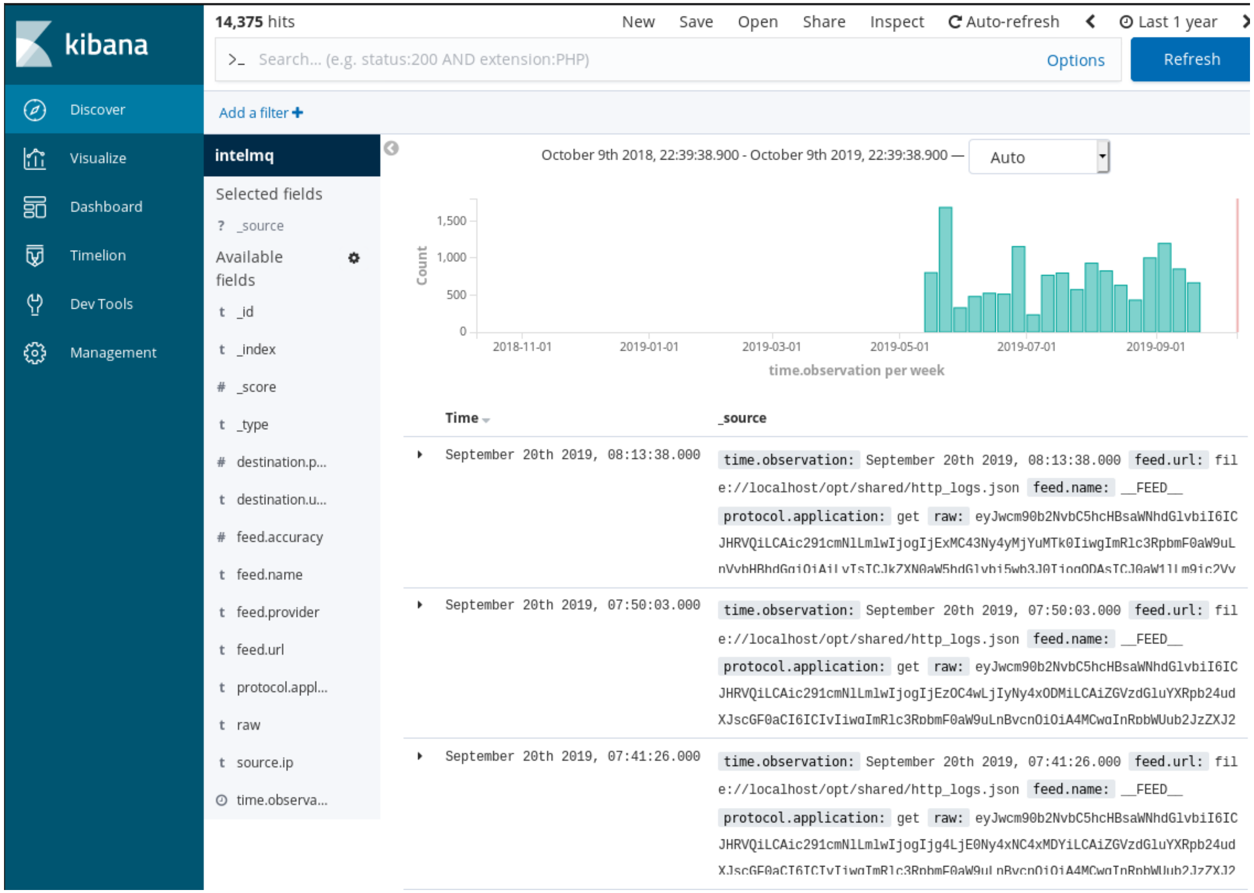
Step 2 of 2: Configure settings

You've defined **intelmq** as your index pattern. Now you can specif

time.observation

The Time Filter will use this field to filter your data by time.
You can choose not to have a time field, but you will not be able to narrow down your data by a time range.

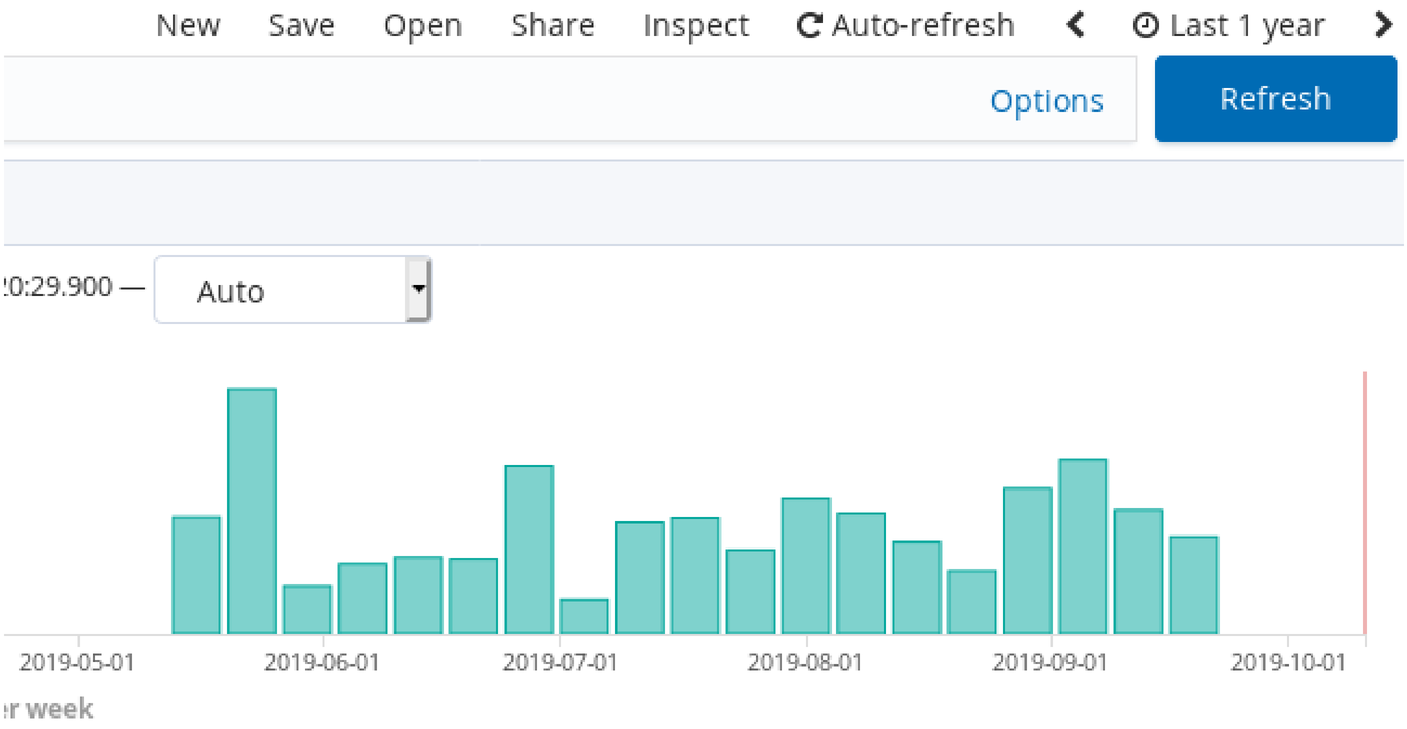
Kibana



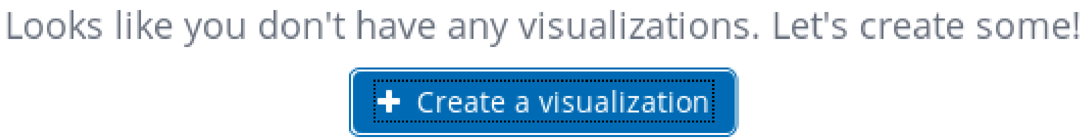
```
September 20th 2019, 08:13:38.000   time.observation: September 20th 2019, 08:13:38.000 feed.url: fil
                                    e://localhost/opt/shared/http_logs.json feed.name: __FEED__
                                    protocol.application: get raw: eyJwcm90b2NvbC5hcHBsaWNhdGlvbiI6IC
                                    JHRVQiLCAic291cmNlLmlwIjogIjExMC43Ny4yMjYuMTk0IiwgImRlc3RpbmF0aW9uL
                                    nVvbHBhdGgiOiAiLyIsICJkZXN0aW5hdGlvbi5wb3J0IjogODAsICJ0aW1lLm9ic2Vy

September 20th 2019, 07:50:03.000   time.observation: September 20th 2019, 07:50:03.000 feed.url: fil
                                    e://localhost/opt/shared/http_logs.json feed.name: __FEED__
                                    protocol.application: get raw: eyJwcm90b2NvbC5hcHBsaWNhdGlvbiI6IC
                                    JHRVQiLCAic291cmNlLmlwIjogIjEzOC4wLjIyNy4xODMiLCAiZGVzdGluYXRpb24ud
                                    XJscGF0aCI6ICIvIiwgImRlc3RpbmF0aW9uLnBvcnQiOiA4MCwgInRpbWUub2JzZXJ2

September 20th 2019, 07:41:26.000   time.observation: September 20th 2019, 07:41:26.000 feed.url: fil
                                    e://localhost/opt/shared/http_logs.json feed.name: __FEED__
                                    protocol.application: get raw: eyJwcm90b2NvbC5hcHBsaWNhdGlvbiI6IC
                                    JHRVQiLCAic291cmNlLmlwIjogIjg4LjE0Ny4xNC4xMDYiLCAiZGVzdGluYXRpb24ud
                                    XJscGF0aCI6ICIvIiwgImRlc3RpbmF0aW9uLnBvcnQiOiA4MCwgInRpbWUub2JzZXJ2
```

Kibana

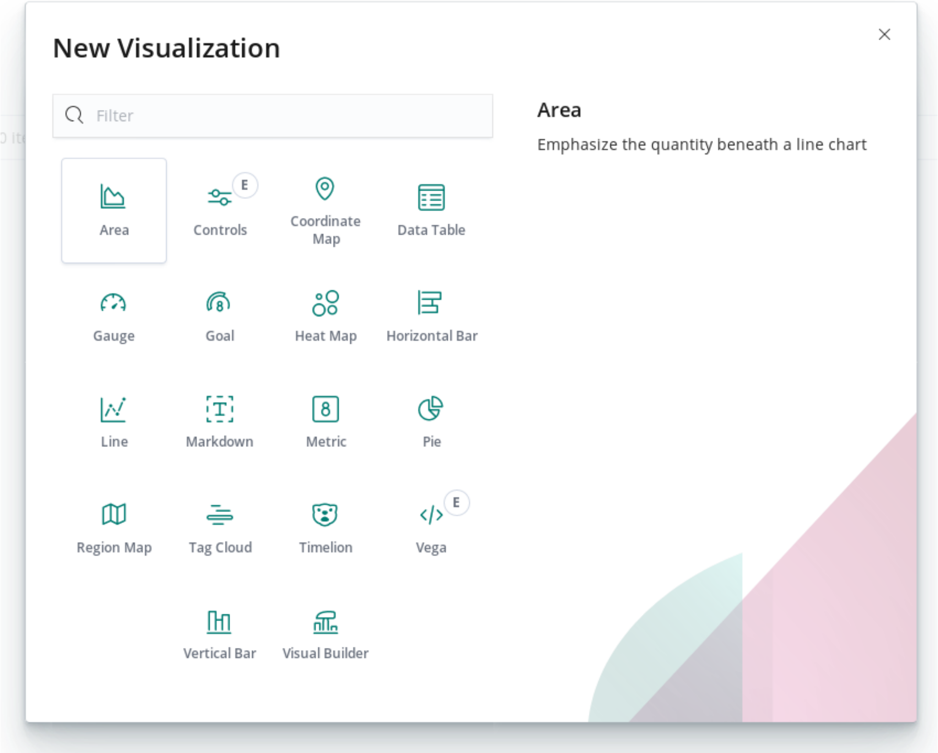
Change timeline range to something much longer, for example 1 year.



Kibana – create visualisation



Looks like you don't have any visualizations. Let's create some!



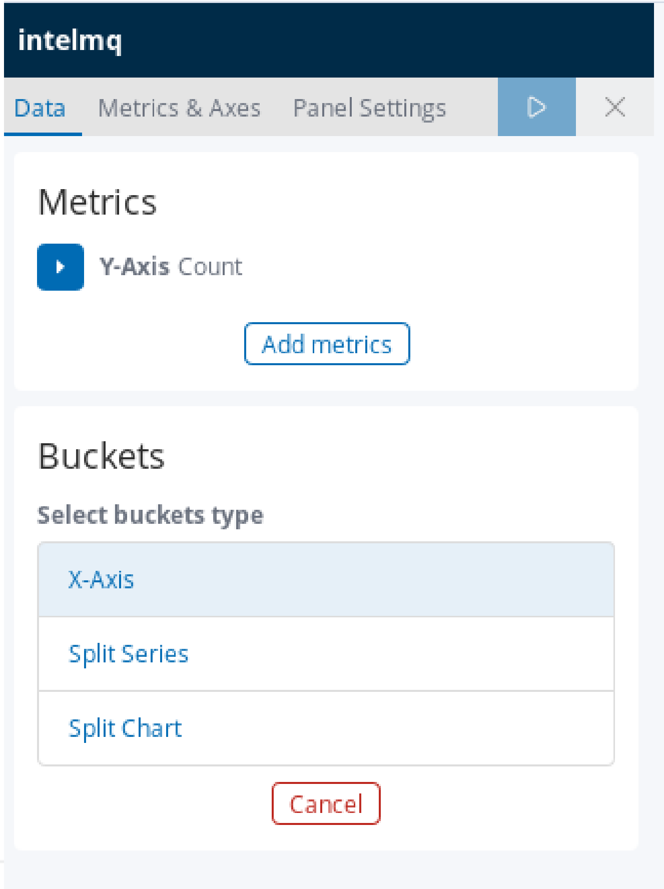
New Visualization

Area

Emphasize the quantity beneath a line chart

Kibana – create visualisation

Pick intelmq as an index (it is the only option) and add a bucket for X axis:



Metrics

Y-Axis Count

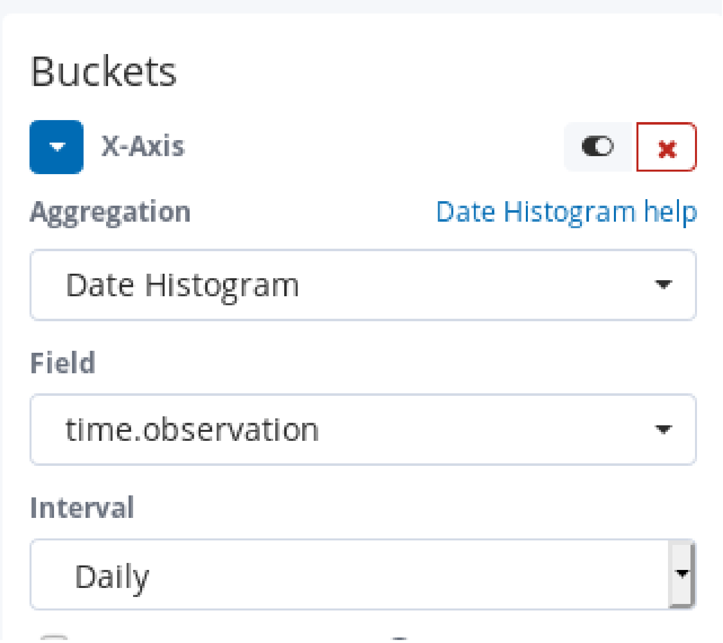
Buckets

Select buckets type

- X-Axis
- Split Series
- Split Chart

Kibana – create visualisation

Date Histogram is a good choice for aggregation, and time.observation is the only available date field. Just pick some reasonable values for interval (for example, Daily or Weekly).



Buckets

X-Axis

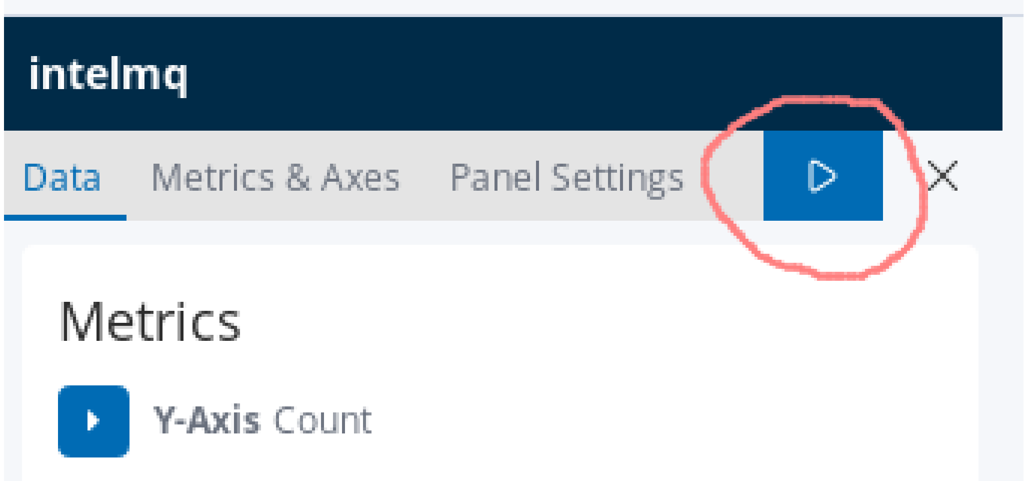
Aggregation: Date Histogram

Field: time.observation

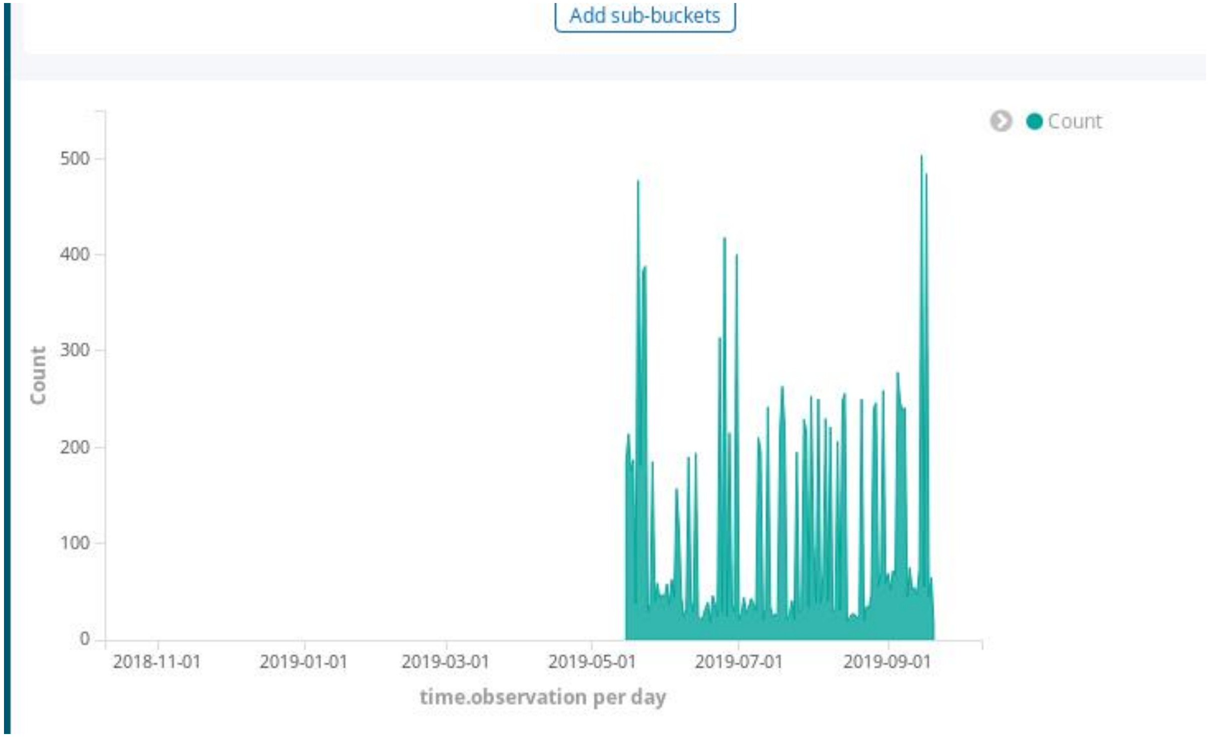
Interval: Daily

Kibana – create visualisation

Confirm with the “play” button above:



Kibana – create visualisation

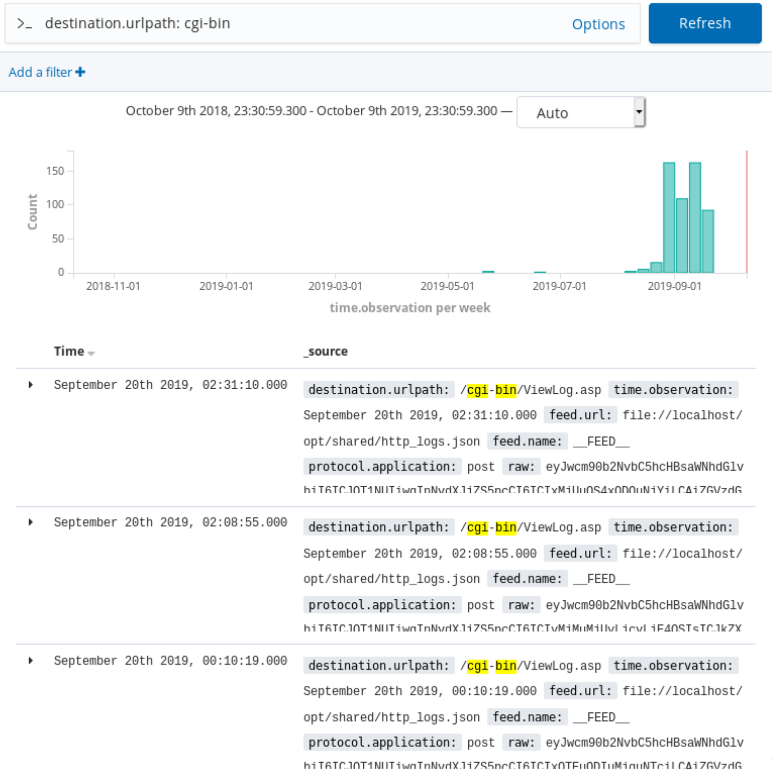


Lucene

- To do a free-text search, just enter a text string. For example: `cgi-bin.`
- To search for a value in a field, enter field name and expected value, separated by colon character. For example: `destination.urlpath: "cgi-bin".`
- Instead of a specific value, you can search for a range of values using bracked squares. It is best explained using an example: `destination.port: [1 TO 1024]`
- You can also combine multiple conditions using AND and OR operators. For example, destination.port: `[1 TO 1024] AND destination.urlpath: "cgi-bin"`.

Lucene

Select Discover in the menu on the left, and type `destination.urlpath: "cgi-bin"` in the big search box on the top. This will allow us to find all url paths with `cgi-bin` as a url component.



October 9th 2018, 23:30:59.300 - October 9th 2019, 23:30:59.300 —

```
▸ September 20th 2019, 02:31:10.000   destination.urlpath: /cgi-bin/ViewLog.asp time.observation:
                                      September 20th 2019, 02:31:10.000 feed.url: file://localhost/
                                      opt/shared/http_logs.json feed.name: __FEED__
                                      protocol.application: post raw: eyJwcm90b2NvbC5hcHBsaWNhdGlv
                                      biI6ICJQT1NUIiwgInNvdXJjZS5pcCI6ICIxMjUuOS4xODQuNjYiLCAiZGVzdG

▸ September 20th 2019, 02:08:55.000   destination.urlpath: /cgi-bin/ViewLog.asp time.observation:
                                      September 20th 2019, 02:08:55.000 feed.url: file://localhost/
                                      opt/shared/http_logs.json feed.name: __FEED__
                                      protocol.application: post raw: eyJwcm90b2NvbC5hcHBsaWNhdGlv
                                      biI6ICJQT1NUIiwgInNvdXJjZS5pcCI6ICIyMjMuMjUyLjcyLjE4OSIsICJkZX

▸ September 20th 2019, 00:10:19.000   destination.urlpath: /cgi-bin/ViewLog.asp time.observation:
                                      September 20th 2019, 00:10:19.000 feed.url: file://localhost/
                                      opt/shared/http_logs.json feed.name: __FEED__
                                      protocol.application: post raw: eyJwcm90b2NvbC5hcHBsaWNhdGlv
                                      biI6ICJQT1NUIiwgInNvdXJjZS5pcCI6ICIxOTEuODIuMjguNTciLCAiZGVzdG
```

IntelMQ Analyst Module

log analysis

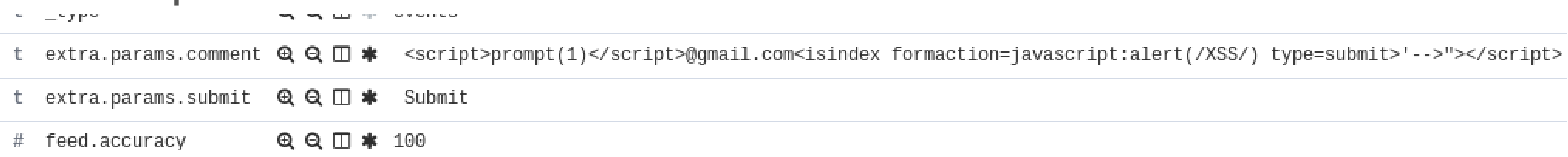
Exercices

Exercise 1

Another commonly exploited endpoint is /wp-admin (wordpress admin interface). Find all requests directed to wp-admin. Are they suspicious? Why?

Exercise 2

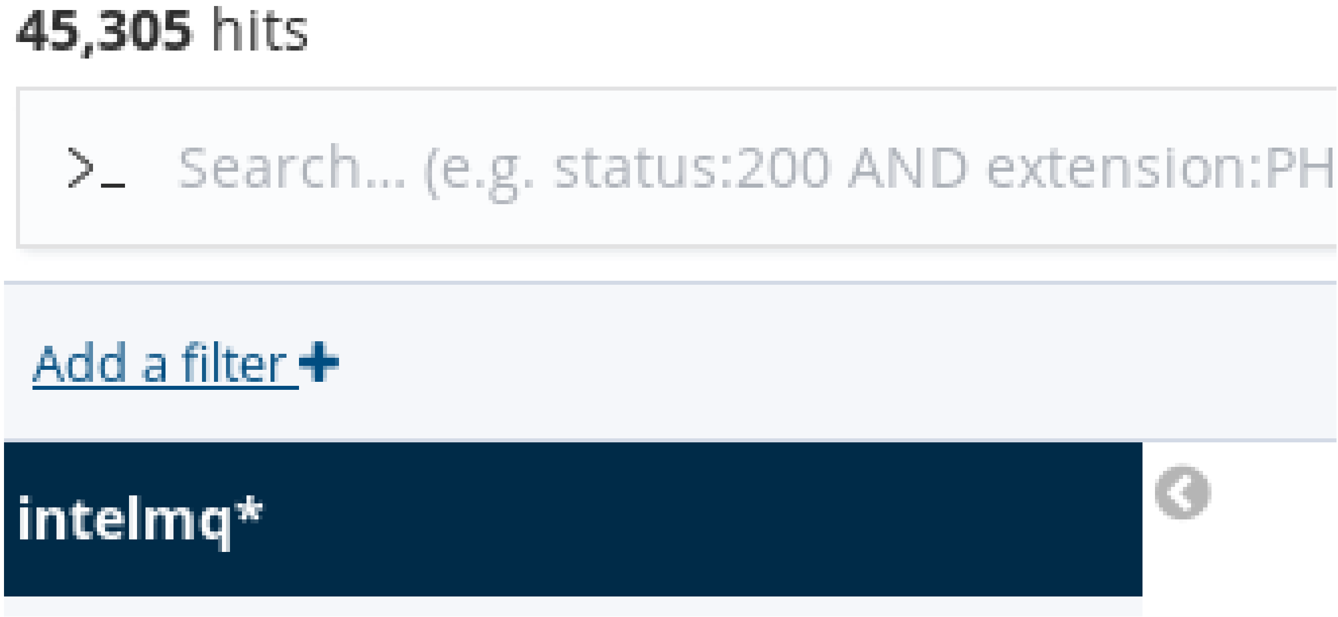
Data from the honeypot looks a bit different. For example, POST and GET parameters are saved:



t	_type	events
t	extra.params.comment	<script>prompt(1)</script>@gmail.com<isindex formaction=javascript:alert(/XSS/) type=submit>'-->"></script>
t	extra.params.submit	Submit
#	feed.accuracy	100

Exercise 2

Filter by requests that have some data submitted. Add a filter, this time using a UI. First, click “Add a filter” button:

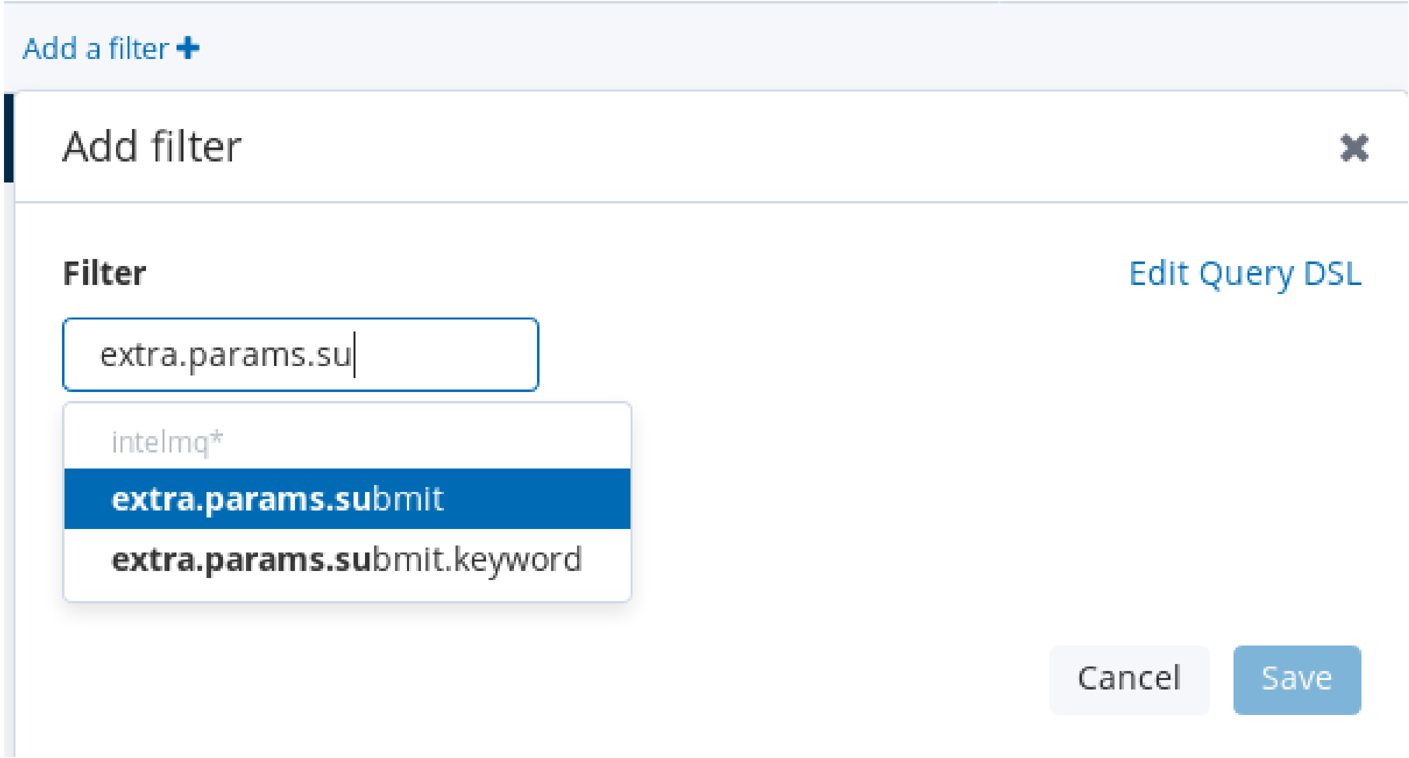


45,305 hits

>_ Search... (e.g. status:200 AND extension:PH

Exercise 2

Type “extra.params.submit”, or select it from the list:



Filter

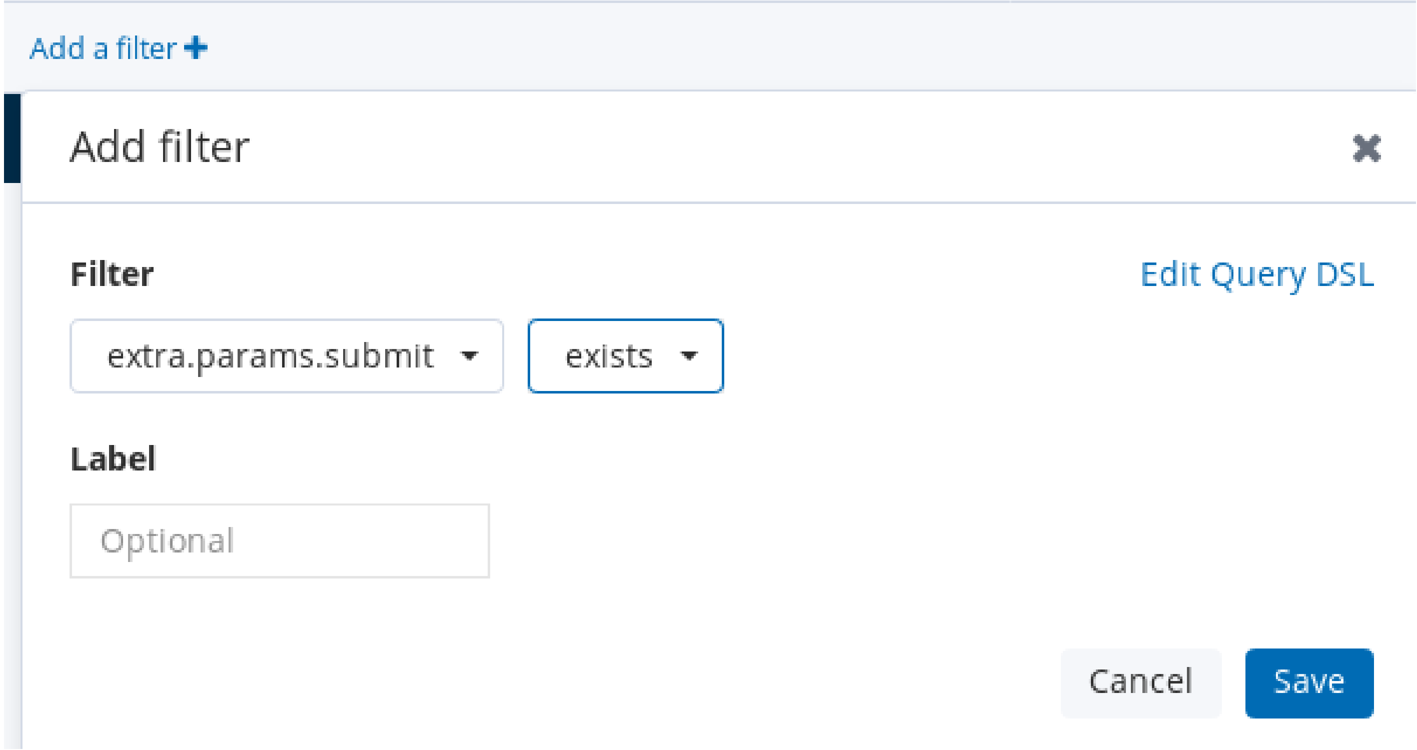
extra.params.su

extra.params.submit

extra.params.submit.keyword

Exercise 2

Pick option “exists”, and click “save”:



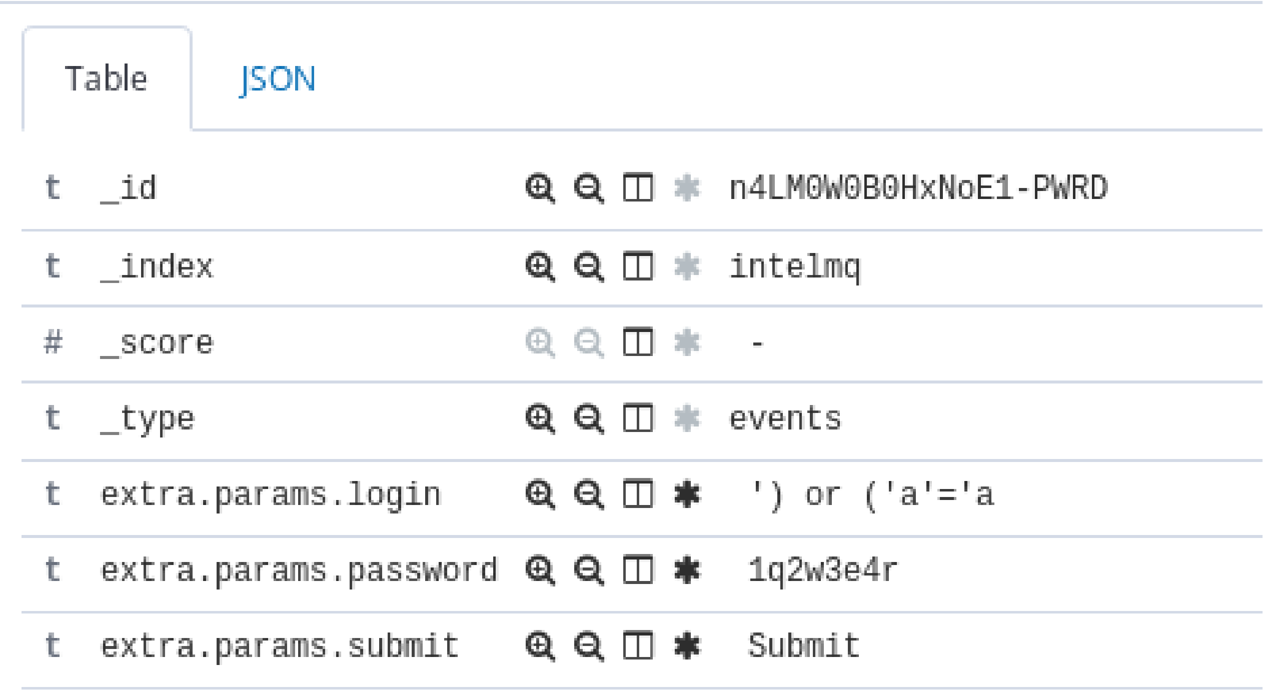
Add a filter ✚

Add filter ✖

Filter: extra.params.submit ▾ exists ▾

Exercise 2

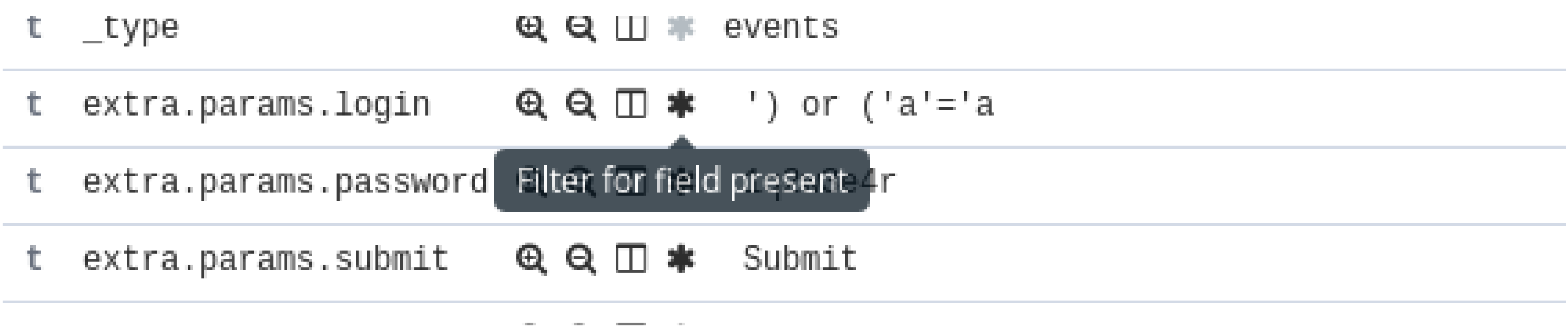
When you browse the results, you will soon find one with a login attempts - like the following



t	_id	n4LM0W0B0HxNoE1-PWRD
t	_index	intelmq
#	_score	-
t	_type	events
t	extra.params.login	') or ('a'='a
t	extra.params.password	1q2w3e4r
t	extra.params.submit	Submit

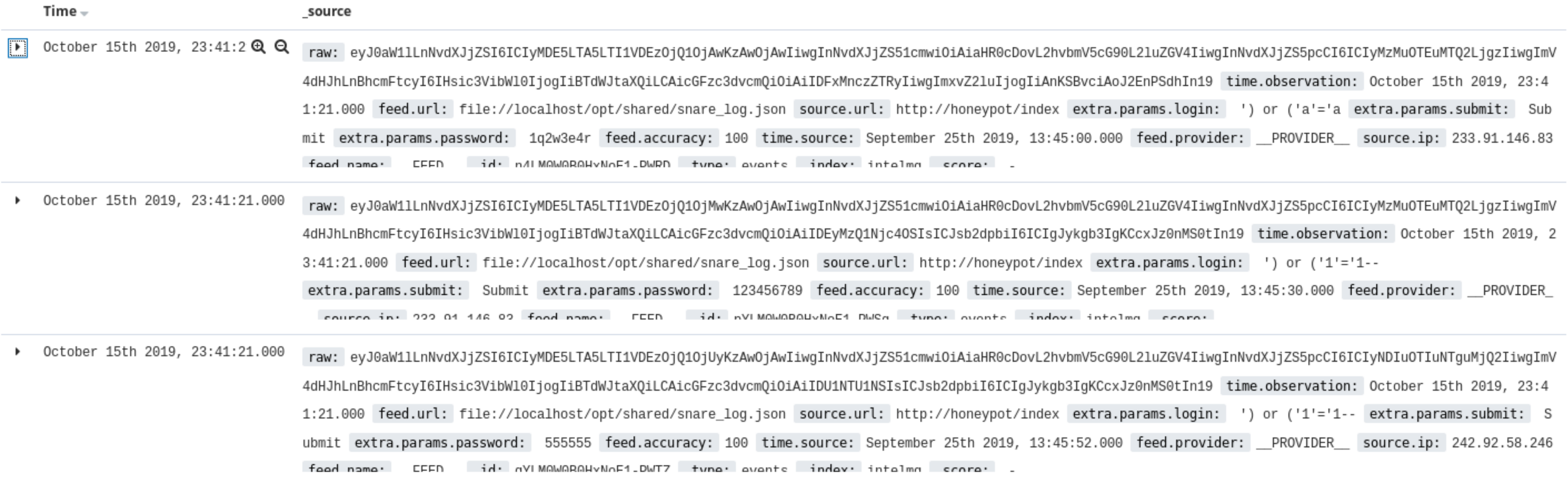
Exercise 2

Let’s filter only for results with extra.params.login. This time we will add a filter even more directly, by clicking a button next to the result



```
t  _type                 events
t  extra.params.login     ') or ('a'='a
t  extra.params.password    4r
t  extra.params.submit    Submit
```

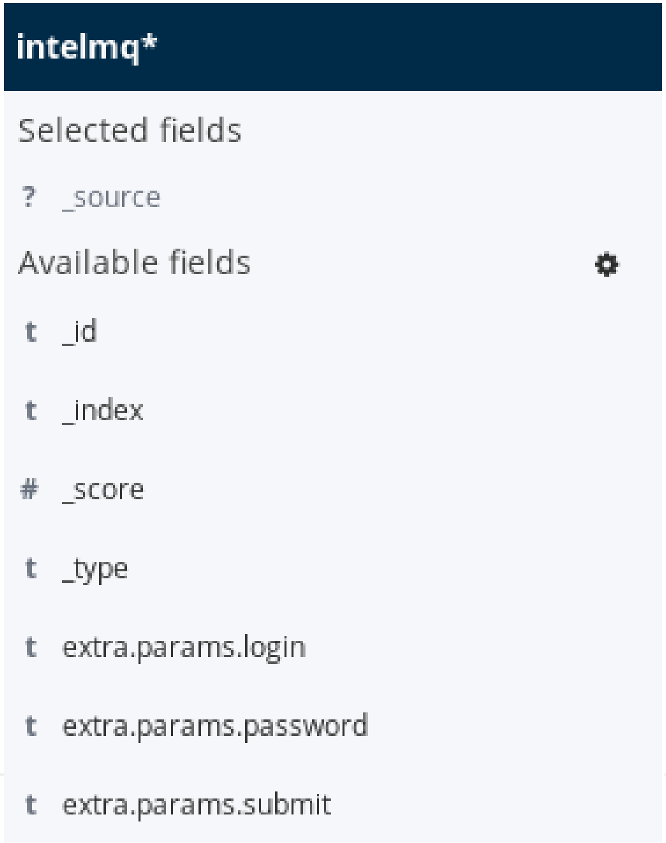
Exercise 2



```
Time      _source

October 15th 2019, 23:41:2   raw: eyJ0aW1lLnNvdXJjZSI6ICIyMDE5LTA5LTI1VDEzOjQ1OjAwKzAwOjAwIiwgInNvdXJjZS51cmwiOiAiaHR0cDovL2hvbmV5cG90L2luZGV4IiwgInNvdXJjZS5pcCI6ICIyMzMuOTEuMTQ2LjgzIiwgImV4dHJhLnBhcmFtcyI6IHsic3VibWl0IjogIiBTdWJtaXQiLCAicGFzc3dvcmQiOiAiIDFxMnczZTRyIiwgImxvZ2luIjogIiAnKSBvciAoJ2EnPSdhIn19 time.observation: October 15th 2019, 23:41:21.000 feed.url: file://localhost/opt/shared/snare_log.json source.url: http://honeypot/index extra.params.login:  ') or ('a'='a extra.params.submit:  Submit extra.params.password:  1q2w3e4r feed.accuracy: 100 time.source: September 25th 2019, 13:45:00.000 feed.provider: __PROVIDER__ source.ip: 233.91.146.83 feed.name:  FEED _id: n4LMQWQBQHxNoE1-PWRD _type: events _index: intelmq _score: -

October 15th 2019, 23:41:21.000   raw: eyJ0aW1lLnNvdXJjZSI6ICIyMDE5LTA5LTI1VDEzOjQ1OjMwKzAwOjAwIiwgInNvdXJjZS51cmwiOiAiaHR0cDovL2hvbmV5cG90L2luZGV4IiwgInNvdXJjZS5pcCI6ICIyMzMuOTEuMTQ2LjgzIiwgImV4dHJhLnBhcmFtcyI6IHsic3VibWl0IjogIiBTdWJtaXQiLCAicGFzc3dvcmQiOiAiIDEyMzQ1Njc4OSIsICJsb2dpbiI6ICIgJykgb3IgKCcxJz0nMS0tIn19 time.observation: October 15th 2019, 23:41:21.000 feed.url: file://localhost/opt/shared/snare_log.json source.url: http://honeypot/index extra.params.login:  ') or ('1'='1-- extra.params.submit:  Submit extra.params.password:  123456789 feed.accuracy: 100 time.source: September 25th 2019, 13:45:30.000 feed.provider: __PROVIDER__ source.ip: 233.91.146.83 feed.name:  FEED _id: nYLMQWQBQHxNoE1_PWSq _type: events _index: intelmq _score:

October 15th 2019, 23:41:21.000   raw: eyJ0aW1lLnNvdXJjZSI6ICIyMDE5LTA5LTI1VDEzOjQ1OjUyKzAwOjAwIiwgInNvdXJjZS51cmwiOiAiaHR0cDovL2hvbmV5cG90L2luZGV4IiwgInNvdXJjZS5pcCI6ICIyNDIuOTIuNTguMjQ2IiwgImV4dHJhLnBhcmFtcyI6IHsic3VibWl0IjogIiBTdWJtaXQiLCAicGFzc3dvcmQiOiAiIDU1NTU1NSIsICJsb2dpbiI6ICIgJykgb3IgKCcxJz0nMS0tIn19 time.observation: October 15th 2019, 23:41:21.000 feed.url: file://localhost/opt/shared/snare_log.json source.url: http://honeypot/index extra.params.login:  ') or ('1'='1-- extra.params.submit:  Submit extra.params.password:  555555 feed.accuracy: 100 time.source: September 25th 2019, 13:45:52.000 feed.provider: __PROVIDER__ source.ip: 242.92.58.246 feed.name:  FEED _id: gYLMQWQBQHxNoE1-PWTZ _type: events _index: intelmq _score: -
```

Exercise 2

To fix this problem, select proper fields in the field selection box and click add. Do this for extra.params.login and extra.params.password:



intelmq*

Selected fields

? _source

Available fields

t _id

t _index

_score

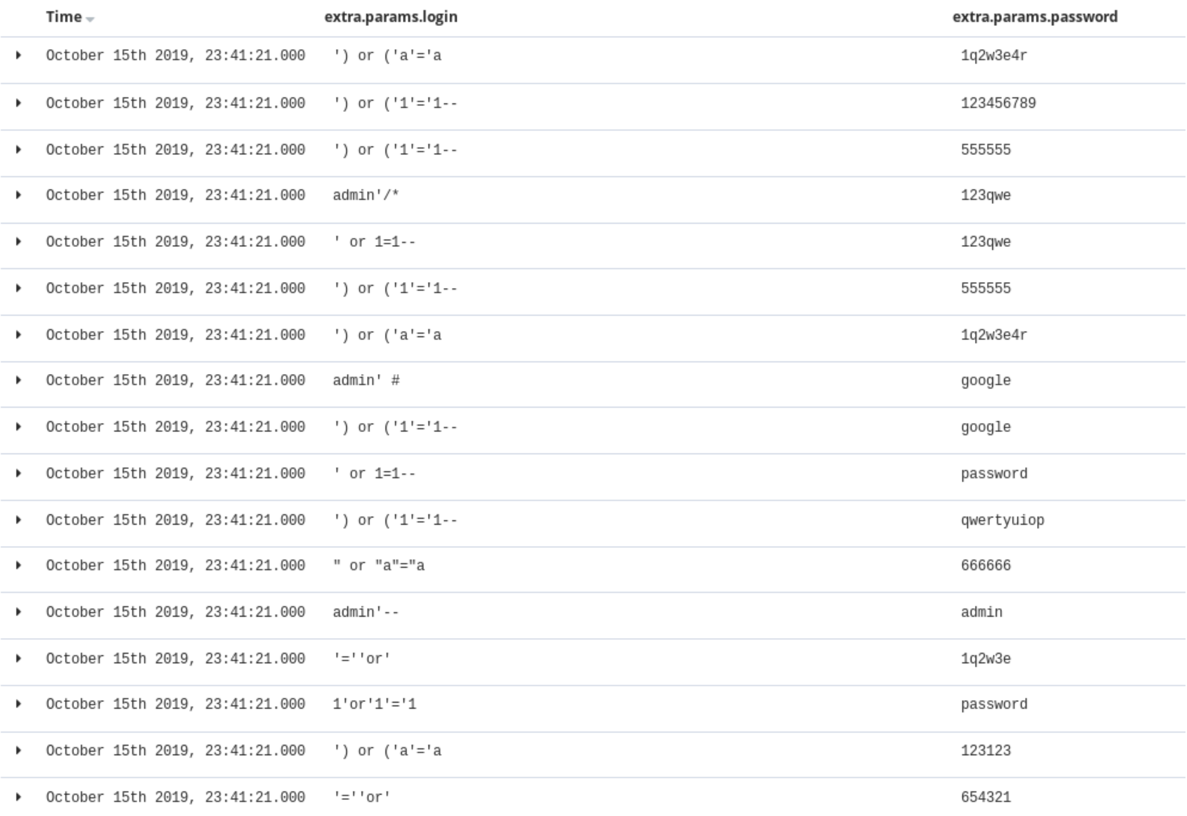
t _type

t extra.params.login

t extra.params.password

t extra.params.submit

Exercise 2



	Time ▾	extra.params.login	extra.params.password
▸	October 15th 2019, 23:41:21.000	') or ('a'='a	1q2w3e4r
▸	October 15th 2019, 23:41:21.000	') or ('1'='1--	123456789
▸	October 15th 2019, 23:41:21.000	') or ('1'='1--	555555
▸	October 15th 2019, 23:41:21.000	admin'/*	123qwe
▸	October 15th 2019, 23:41:21.000	' or 1=1--	123qwe
▸	October 15th 2019, 23:41:21.000	') or ('1'='1--	555555
▸	October 15th 2019, 23:41:21.000	') or ('a'='a	1q2w3e4r
▸	October 15th 2019, 23:41:21.000	admin' #	google
▸	October 15th 2019, 23:41:21.000	') or ('1'='1--	google
▸	October 15th 2019, 23:41:21.000	' or 1=1--	password
▸	October 15th 2019, 23:41:21.000	') or ('1'='1--	qwertyuiop
▸	October 15th 2019, 23:41:21.000	" or "a"="a	666666
▸	October 15th 2019, 23:41:21.000	admin'--	admin
▸	October 15th 2019, 23:41:21.000	'=''or'	1q2w3e
▸	October 15th 2019, 23:41:21.000	1'or'1'='1	password
▸	October 15th 2019, 23:41:21.000	') or ('a'='a	123123
▸	October 15th 2019, 23:41:21.000	'=''or'	654321

Exercise 2

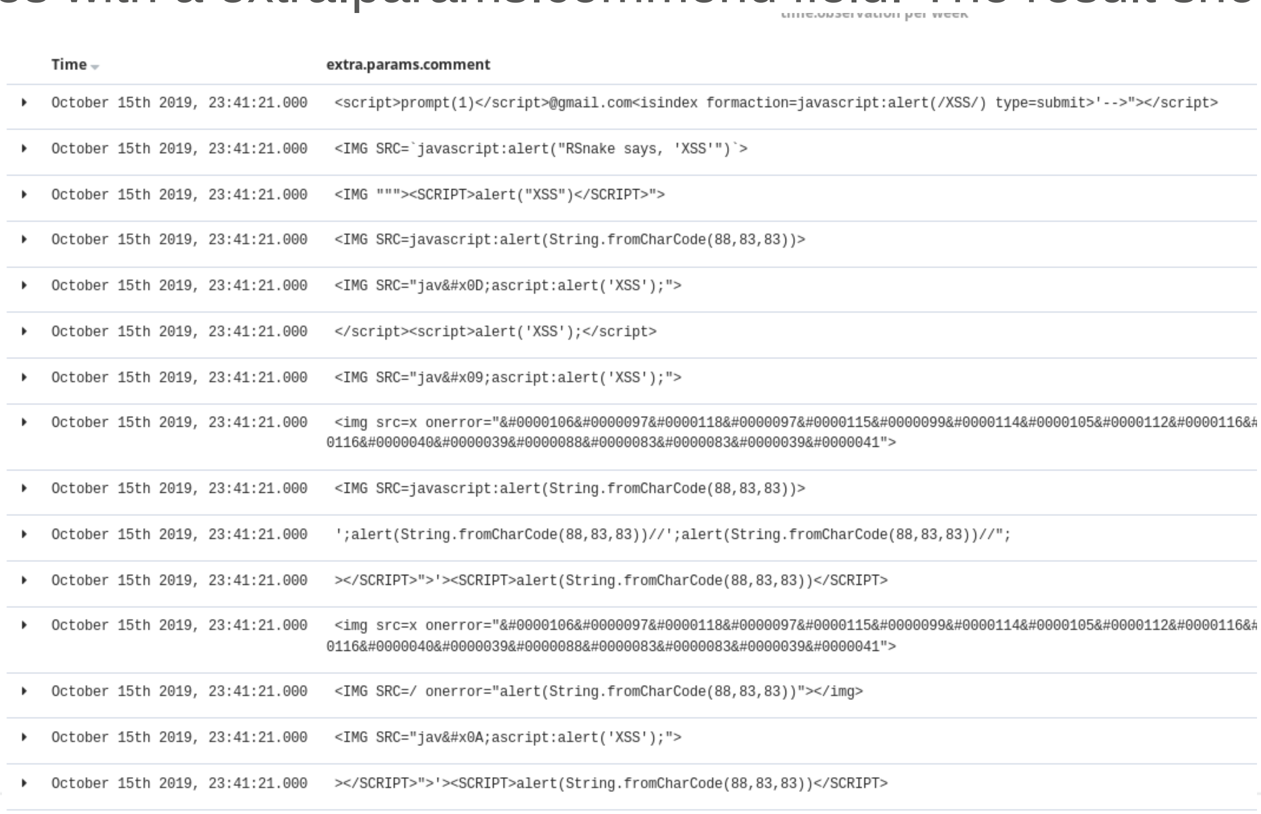
Can you tell what kind of attack against the webaplication is attempted here (hint - it is one of OWASP top10 attacks)? What are the countermeasures against this attack? What are the possible repercussions?

Find a few most commonly attempted passwords. Are they strong or weak on average? Do you think that a company policy with a blacklist of forbidden passwords is a good idea? If yes, which freely available data sources or APIs would you use to get a better list of easily crackable passwords?

Prepare a short advisory for your constituency. It should contain a warning against this kind of attacks, and specific details for this campaign, including a list of most common attempted passwords.

Exercise 3

Add a field extra.params.commend and add a filter to select only messages with a extra.params.commend field. The result should look like this:



time.observation per week

Time	extra.params.comment
October 15th 2019, 23:41:21.000	<script>prompt(1)</script>@gmail.com<isindex formaction=javascript:alert(/XSS/) type=submit>'-->"></script>
October 15th 2019, 23:41:21.000	<IMG SRC=`javascript:alert("RSnake says, 'XSS'")`>
October 15th 2019, 23:41:21.000	<IMG """><SCRIPT>alert("XSS")</SCRIPT>">
October 15th 2019, 23:41:21.000	<IMG SRC=javascript:alert(String.fromCharCode(88,83,83))>
October 15th 2019, 23:41:21.000	<IMG SRC="jav&#x0D;ascript:alert('XSS');">
October 15th 2019, 23:41:21.000	</script><script>alert('XSS');</script>
October 15th 2019, 23:41:21.000	<IMG SRC="jav&#x09;ascript:alert('XSS');">
October 15th 2019, 23:41:21.000	<img src=x onerror="&#0000106&#0000097&#0000118&#0000097&#0000115&#0000099&#0000114&#0000105&#0000112&#0000116&# 0116&#0000040&#0000039&#0000088&#0000083&#0000083&#0000039&#0000041">
October 15th 2019, 23:41:21.000	<IMG SRC=javascript:alert(String.fromCharCode(88,83,83))>
October 15th 2019, 23:41:21.000	';alert(String.fromCharCode(88,83,83))//';alert(String.fromCharCode(88,83,83))//";
October 15th 2019, 23:41:21.000	></SCRIPT>">'><SCRIPT>alert(String.fromCharCode(88,83,83))</SCRIPT>
October 15th 2019, 23:41:21.000	<img src=x onerror="&#0000106&#0000097&#0000118&#0000097&#0000115&#0000099&#0000114&#0000105&#0000112&#0000116&# 0116&#0000040&#0000039&#0000088&#0000083&#0000083&#0000039&#0000041">
October 15th 2019, 23:41:21.000	<IMG SRC=/ onerror="alert(String.fromCharCode(88,83,83))"></img>
October 15th 2019, 23:41:21.000	<IMG SRC="jav&#x0A;ascript:alert('XSS');">
October 15th 2019, 23:41:21.000	></SCRIPT>">'><SCRIPT>alert(String.fromCharCode(88,83,83))</SCRIPT>

Exercise 3

Can you tell what kind of attack against the webaplication is attempted here (hint - it is one of OWASP top10 attacks)? What are the countermeasures against this attack? What are the possible repercussions?

Most attacks have only local code, but some exploit attempts are referencing an external server. Find URLs of the external servers used in the atack.

Prepare a short advisory for your constituency. It should contain a warning against this kind of attacks, and specific details for this campaign, including a list of servers used by the attackers.

CHAPTER 6

ORCHESTRATION OF CSIRT TOOLS

THEHIVE ANALYST MODULE

ACTING ON THREAT INTELLIGENCE

AGENDA

- Introduction to exercise
- Task 1: Understanding general workflow of TheHive
- Task 2: Getting familiar with TheHive interface
- Task 3: Performing an investigation of provided case by creating tasks, enriching data using Cortex analyzers and discussion on obtained results.

TheHive Analyst Module

acting on threat intelligence

Introduction

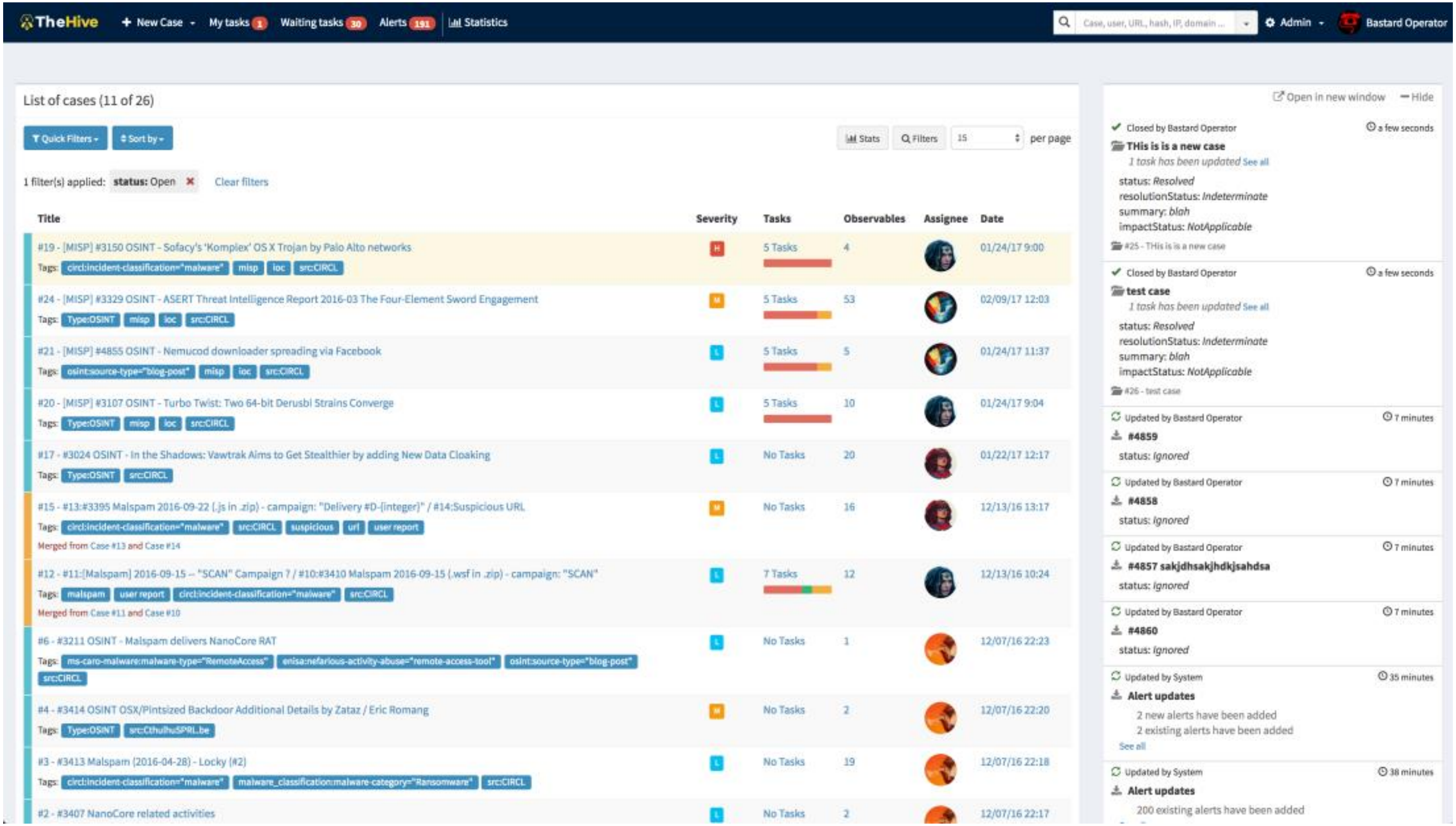
Why TheHive?

- System is dedicated for Security Operational Centers
- Easy way to conduct investigations
- Many users can work in parallel
- Useful built-in tools for data enrichment
- Autocorrelation of tags and observables
- Noncomplicated integration with MISP

TheHive Alternatives

- **Maltego (Commercial)**
  - https://www.paterva.com/buy/maltego-clients.php
- **FIR - Fast Incident Response**
  - https://github.com/certsocietegenerale/FIR

TheHive main view



List of cases (11 of 26)

1 filter(s) applied: **status:** Open ✖ Clear filters

Title	Severity	Tasks	Observables	Assignee	Date
#19 - [MISP] #3150 OSINT - Sofacy's 'Komplex' OS X Trojan by Palo Alto networks Tags: circl:incident-classification="malware" misp ioc src:CIRCL	H	5 Tasks	4		01/24/17 9:00
#24 - [MISP] #3329 OSINT - ASERT Threat Intelligence Report 2016-03 The Four-Element Sword Engagement Tags: Type:OSINT misp ioc src:CIRCL	M	5 Tasks	53		02/09/17 12:03
#21 - [MISP] #4855 OSINT - Nemucod downloader spreading via Facebook Tags: osint:source-type="blog-post" misp ioc src:CIRCL	L	5 Tasks	5		01/24/17 11:37
#20 - [MISP] #3107 OSINT - Turbo Twist: Two 64-bit Derusbi Strains Converge Tags: Type:OSINT misp ioc src:CIRCL	L	5 Tasks	10		01/24/17 9:04
#17 - #3024 OSINT - In the Shadows: Vawtrak Aims to Get Stealthier by adding New Data Cloaking Tags: Type:OSINT src:CIRCL	L	No Tasks	20		01/22/17 12:17
#15 - #13:#3395 Malspam 2016-09-22 (.js in .zip) - campaign: "Delivery #D-(integer)" / #14:Suspicious URL Tags: circl:incident-classification="malware" src:CIRCL suspicious url user report Merged from Case #13 and Case #14	M	No Tasks	16		12/13/16 13:17
#12 - #11:[Malspam] 2016-09-15 -- "SCAN" Campaign 7 / #10:#3410 Malspam 2016-09-15 (.wsf in .zip) - campaign: "SCAN" Tags: malspam user report circl:incident-classification="malware" src:CIRCL Merged from Case #11 and Case #10	L	7 Tasks	12		12/13/16 10:24
#6 - #3211 OSINT - Malspam delivers NanoCore RAT Tags: ms-caro-malware:malware-type="RemoteAccess" enisa:nefarious-activity-abuse="remote-access-tool" osint:source-type="blog-post" src:CIRCL	L	No Tasks	1		12/07/16 22:23
#4 - #3414 OSINT OSX/Pintsized Backdoor Additional Details by Zataz / Eric Romang Tags: Type:OSINT src:CthulhuSPRL.be	M	No Tasks	2		12/07/16 22:20
#3 - #3413 Malspam (2016-04-28) - Locky (#2) Tags: circl:incident-classification="malware" malware_classification:malware-category="Ransomware" src:CIRCL	L	No Tasks	19		12/07/16 22:18
#2 - #3407 NanoCore related activities	L	No Tasks	2		12/07/16 22:17

✓ Closed by Bastard Operator — a few seconds
THis is is a new case
1 task has been updated See all
status: *Resolved*
resolutionStatus: *Indeterminate*
summary: *blah*
impactStatus: *NotApplicable*
#25 - THis is is a new case

✓ Closed by Bastard Operator — a few seconds
test case
1 task has been updated See all
status: *Resolved*
resolutionStatus: *Indeterminate*
summary: *blah*
impactStatus: *NotApplicable*
#26 - test case

Updated by Bastard Operator — 7 minutes
#4859
status: *Ignored*

Updated by Bastard Operator — 7 minutes
#4858
status: *Ignored*

Updated by Bastard Operator — 7 minutes
#4857 sakjdhsakjhdkjsahdsa
status: *Ignored*

Updated by Bastard Operator — 7 minutes
#4860
status: *Ignored*

Updated by System — 35 minutes
Alert updates
2 new alerts have been added
2 existing alerts have been added
See all

Updated by System — 38 minutes
Alert updates
200 existing alerts have been added

Basic concepts

- **Case - root object of investigation**
- **Task - belongs to Case**
- **Observables - added during the investigation, similar to MISP attributes, can be marked as Indicators of Compromise**
- **Alerts - events can be imported eg. from MISP**

Basic concepts

Each observable must have:

- **TLP**
- **Tag, Description - or both**

Observables can be:

- **domain, IP, hash, file, url ... etc**
- **Flagged as IoC**
- **Tagged**
- **Exported as: csv, text or MISP compatible format**
- **Analyzed via Cortex Analyzers**
- **exported to MISP**

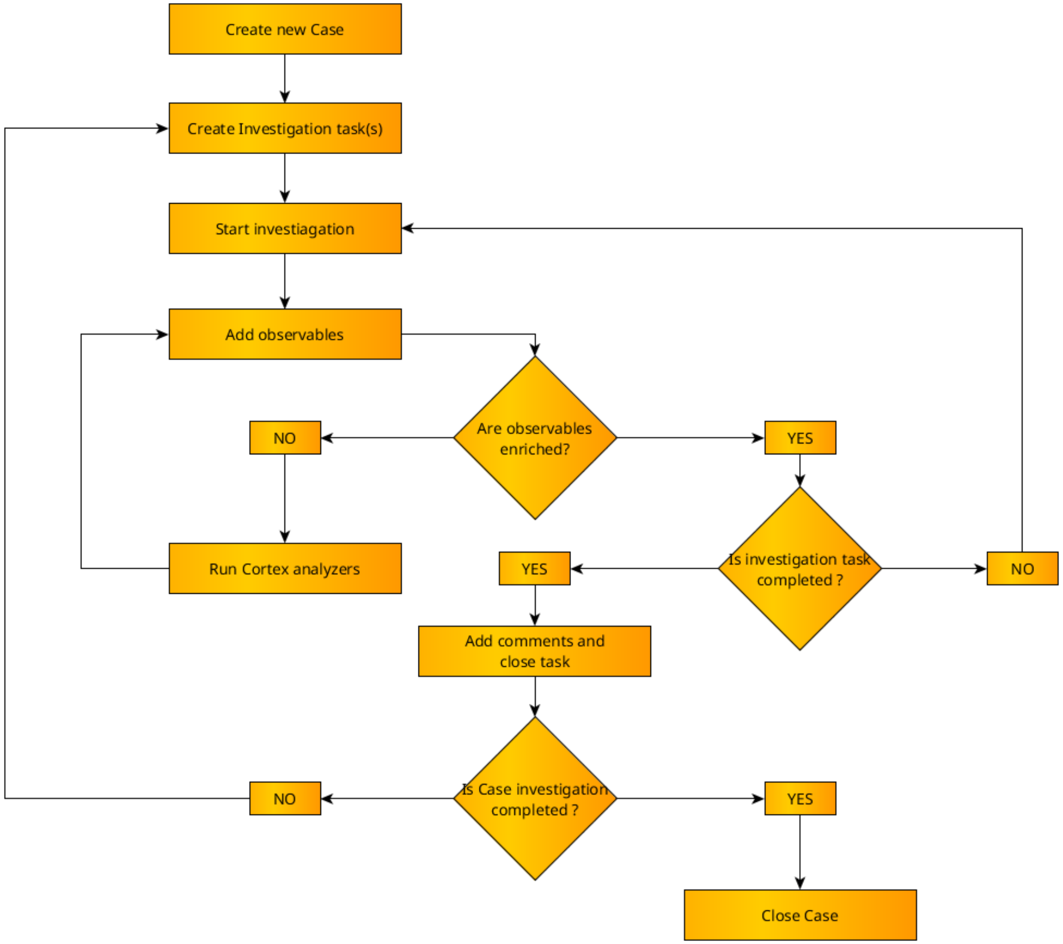
Cortex

- **Cortex - environment for applications called Analyzers**
- **Analyzers can be invoked from TheHive, directly from Cortex web interface, Cortex REST API or Cortex4py**
- **Analyzers output can be customized by templates**
- **Cortex engine has many built-in analyzers written in python**
- **Any programming language can be used for writing analyzer**
- **Easy to write own analyzers**
  - Definition: new_analyzer.json
  - Main script: new_analyzer.py
  - Optionally: requirements.txt

Short description of few built-in analyzers

- **CIRCLPassiveDNS: Check CIRCL's Passive DNS for a given domain.**
- **GoogleSafebrowsing: check URLs against Google Safebrowsing.**
- **MaxMind: geolocation.**
- **MISP Search: search for MISP events in one or several MISP instances containing the observable submitted as input.**
- **VirusTotal: look up files, URLs and hashes through VirusTotal.**
- **Yara: check files against YARA rules using yara-python.**

TheHive Workflow


Create new Case
Create Investigation task(s)
Start investiagation
Add observables
Are observables enriched?
NO
YES
Run Cortex analyzers
Is investigation task completed ?
YES
NO
Add comments and close task
Is Case investigation completed ?
NO
YES
Close Case

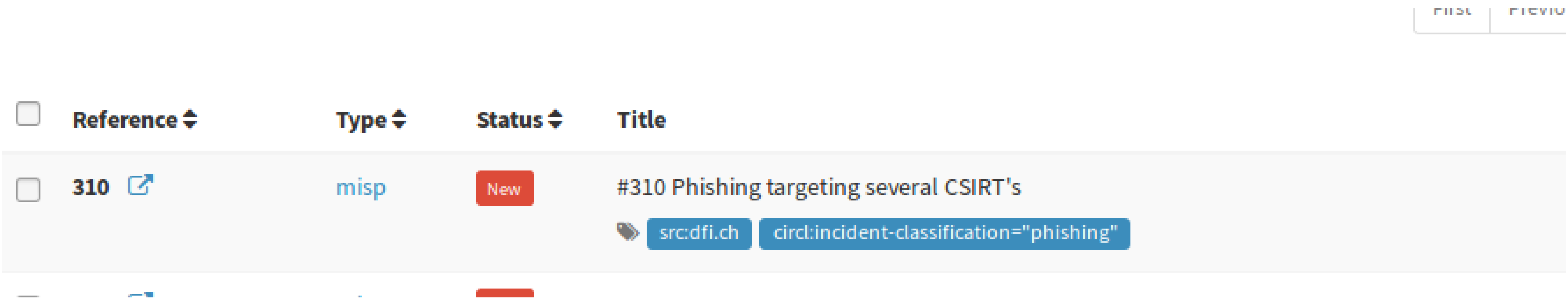
TheHive Analyst Module

acting on threat intelligence

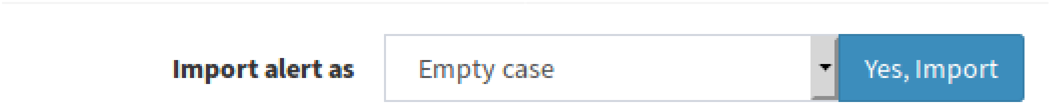
Investigation

Investigation

Find interesting event in Alerts tab



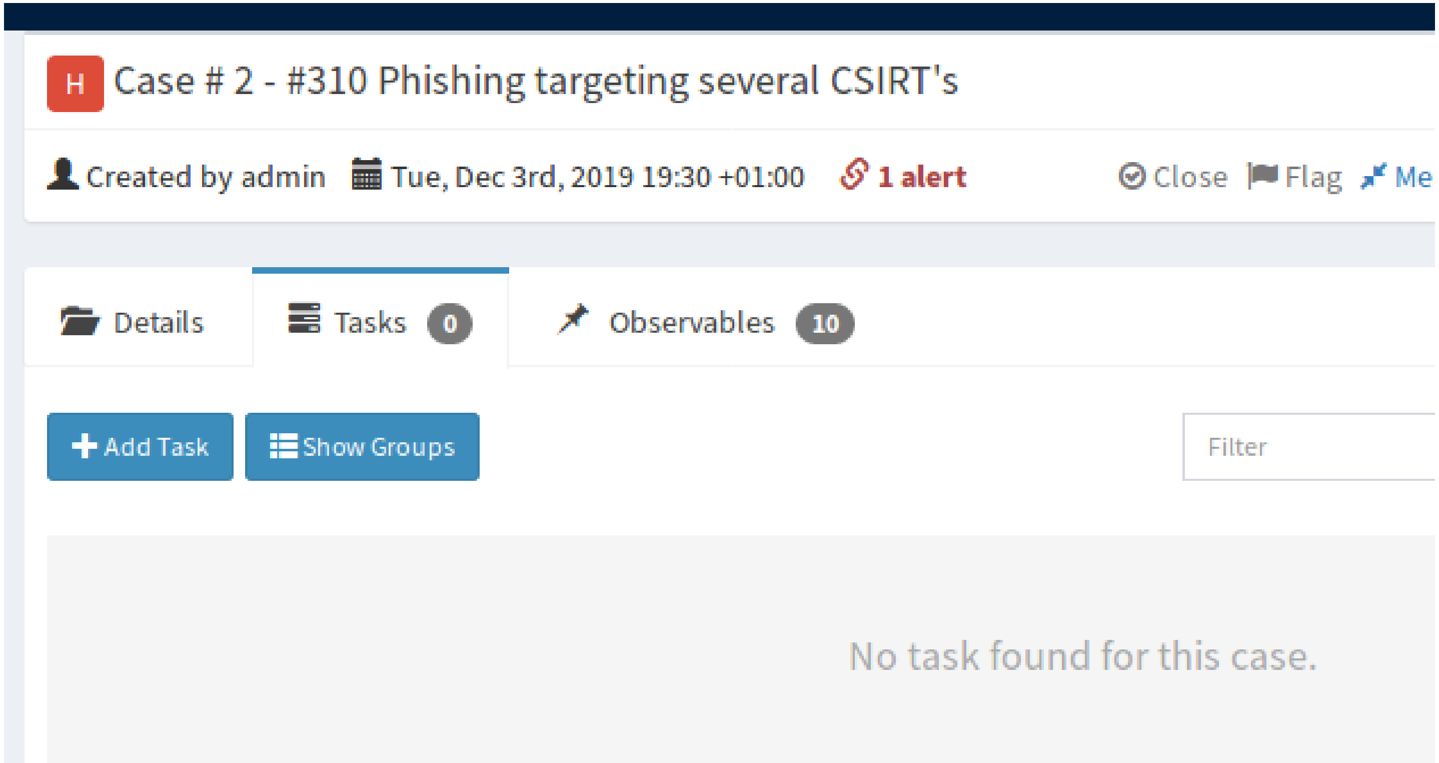
□	Reference ⬍	Type ⬍	Status ⬍	Title
□	310	misp	New	#310 Phishing targeting several CSIRT's src:dfi.ch circl:incident-classification="phishing"



Import alert as: Empty case

Investigation

Create tasks



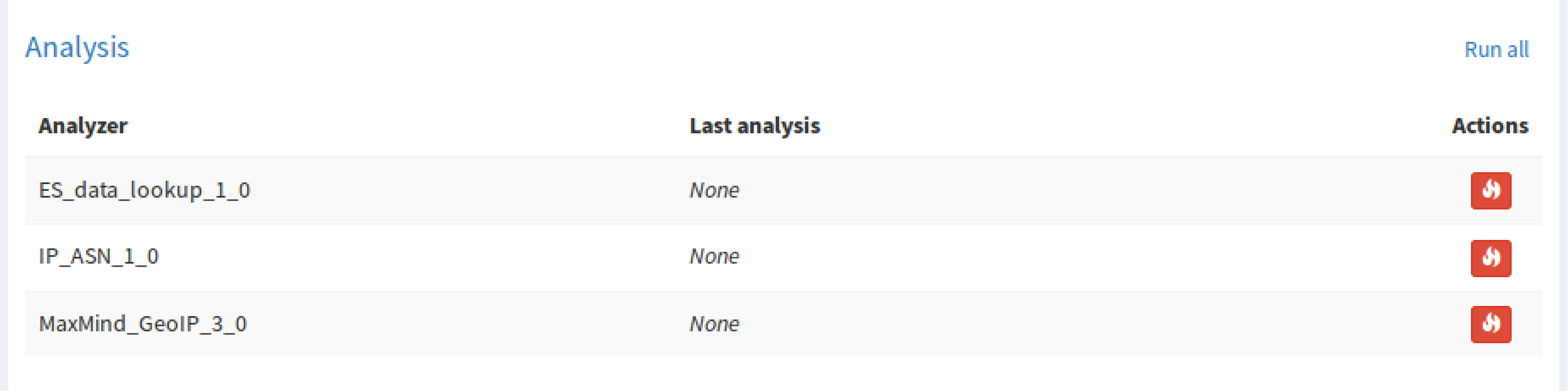
H Case # 2 - #310 Phishing targeting several CSIRT's

Created by admin Tue, Dec 3rd, 2019 19:30 +01:00 1 alert

No task found for this case.

Investigation

Perform the tasks by running analyzers for both IP's

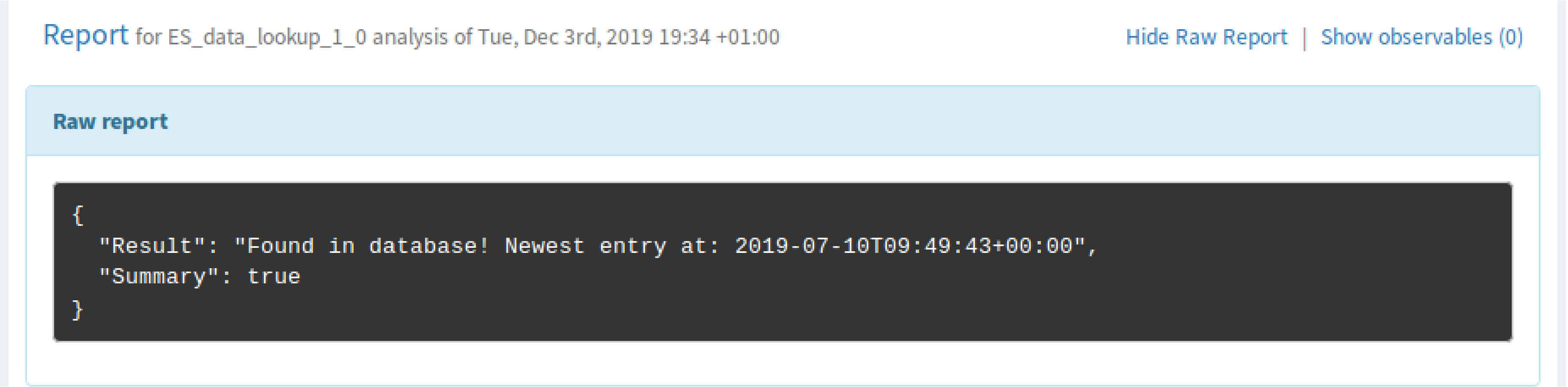


Analysis

Analyzer	Last analysis	Actions
ES_data_lookup_1_0	*None*	🔥
IP_ASN_1_0	*None*	🔥
MaxMind_GeoIP_3_0	*None*	🔥

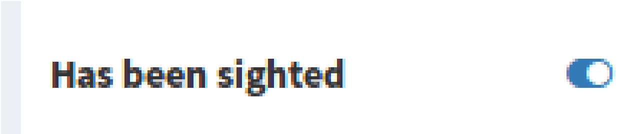
Investigation

If IP was seen in logs, set sighted flag



Report for ES_data_lookup_1_0 analysis of Tue, Dec 3rd, 2019 19:34 +01:00

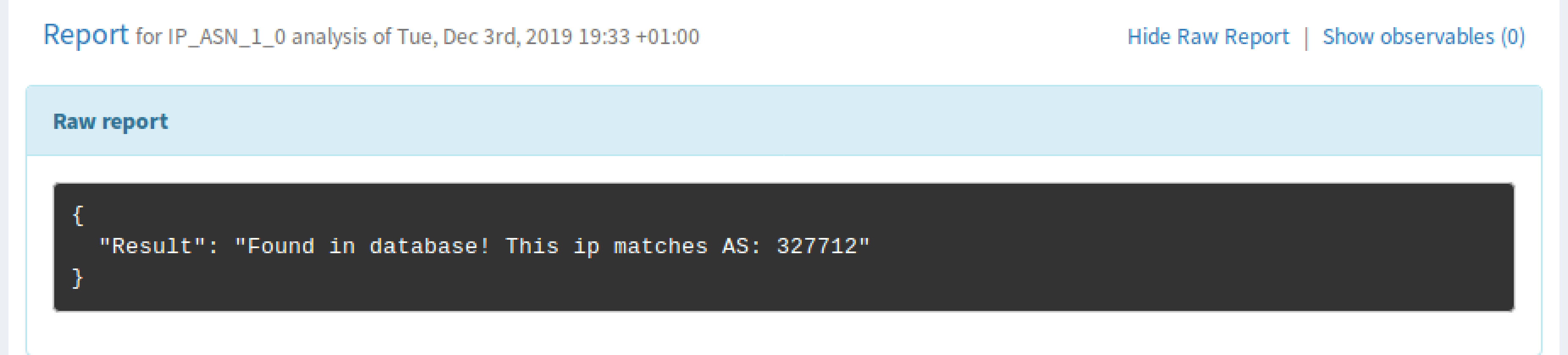
```
{
  "Result": "Found in database! Newest entry at: 2019-07-10T09:49:43+00:00",
  "Summary": true
}
```



Has been sighted

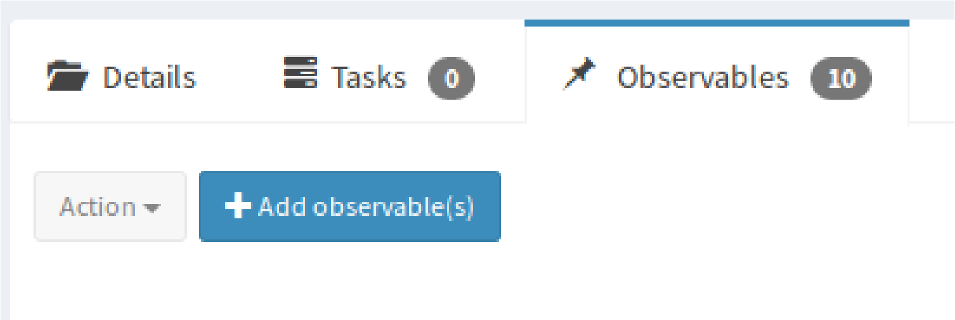
Investigation

If you encounter new data, add them as observable



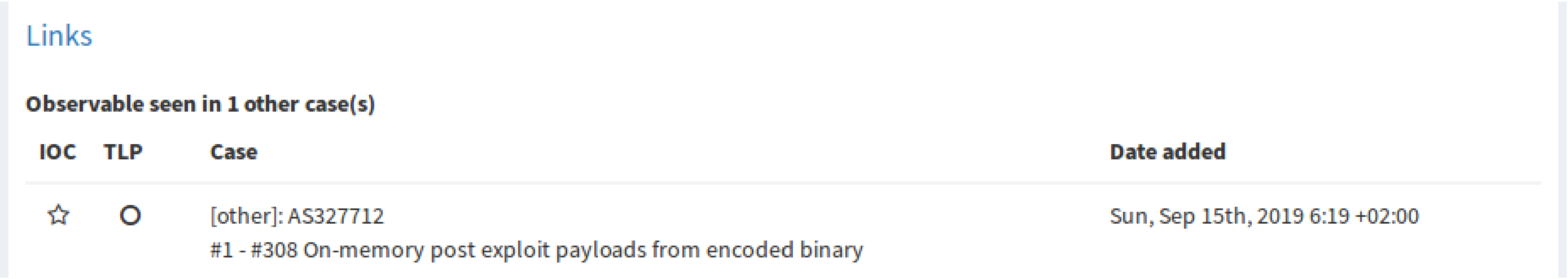
Report for IP_ASN_1_0 analysis of Tue, Dec 3rd, 2019 19:33 +01:00

```
{
  "Result": "Found in database! This ip matches AS: 327712"
}
```



Investigation

Check if you can pivot on added observables



Links

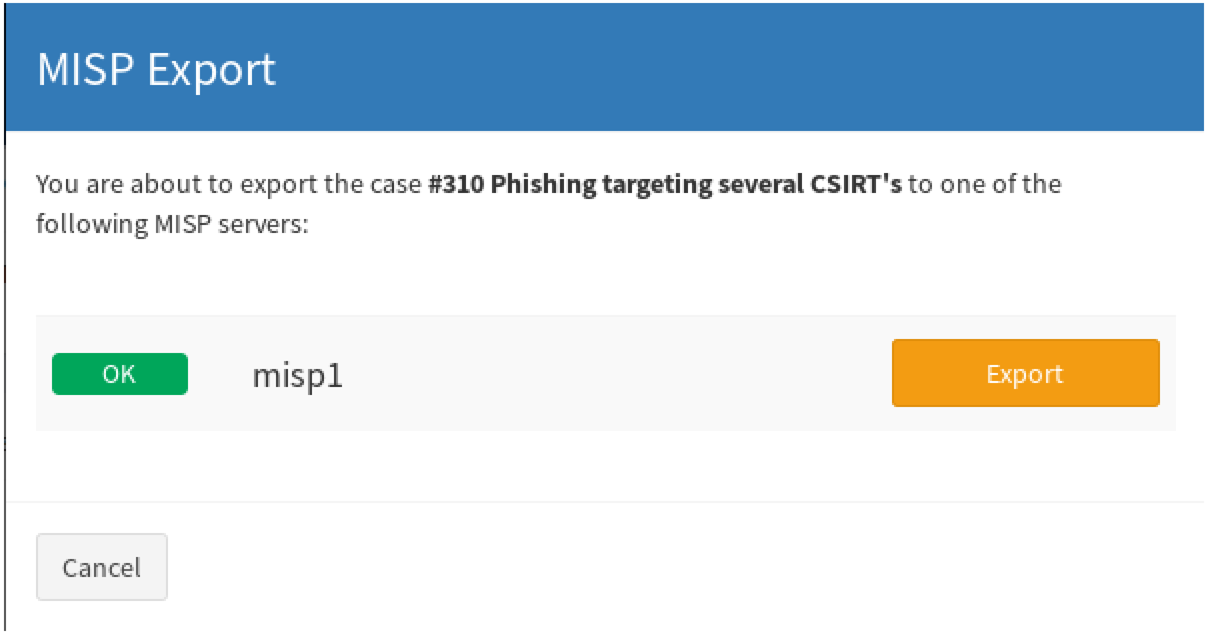
Observable seen in 1 other case(s)

IOC	TLP	Case	Date added
☆	○	[other]: AS327712 #1 - #308 On-memory post exploit payloads from encoded binary	Sun, Sep 15th, 2019 6:19 +02:00

Why is it important?

Investigation

Conclude the investigation by exporting IoCs to MISP



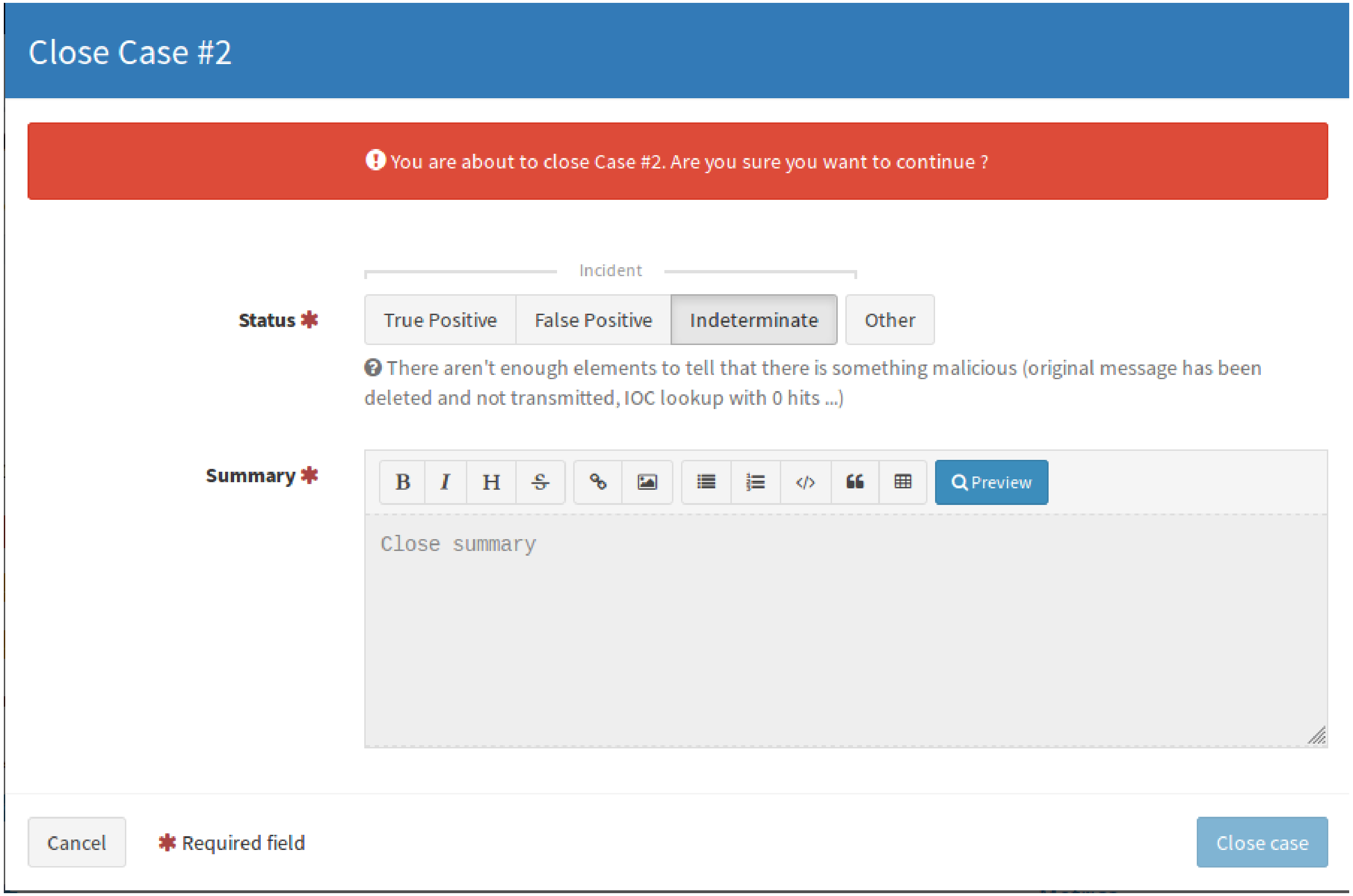
MISP Export

You are about to export the case **#310 Phishing targeting several CSIRT's** to one of the following MISP servers:

Why is it important?

Investigation

If you’re done, close the case.



Close Case #2

You are about to close Case #2. Are you sure you want to continue ?

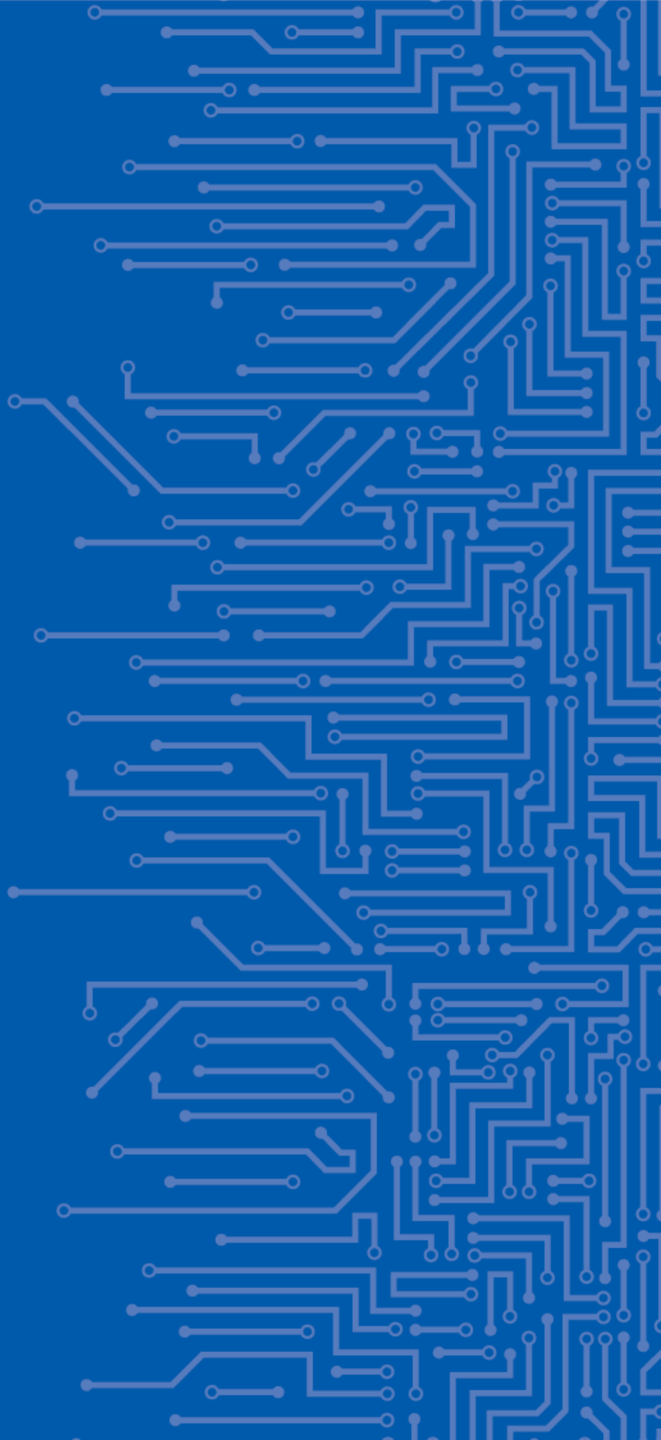
Status ✱ True Positive | False Positive | [✓ Indeterminate] | Other

There aren't enough elements to tell that there is something malicious (original message has been deleted and not transmitted, IOC lookup with 0 hits ...)

Summary ✱

✱ Required field

CHAPTER 7



ORCHESTRATION OF CSIRT TOOLS

TRAINING PLATFORM TECHNOLOGY BACKGROUND

MAIN REQUIREMENTS FOR THE TRAINING PLATFORM

- Must be easy to get up and running
- Must be easy to maintain over time
- Must allow some persistency for multiple initial states
- Should be transferable to a cloud based environment

TECHNOLOGIES USED

- **Kubernetes**

open-source container-orchestration system for automating application deployment, scaling, and management

- **MicroK8s**

packaged version of Kubernetes dedicated for small development deployments

- **Helm**

package manager for Kubernetes that allows developers and operators to more easily package, configure, and deploy applications and services onto Kubernetes clusters

CORE TECHNOLOGIES & CONCEPTS OF THE TRAINING PLATFORM

- **Kubernetes**
  - Container
  - Pod
  - Deployment
  - Service
  - Ingress
  - Volume

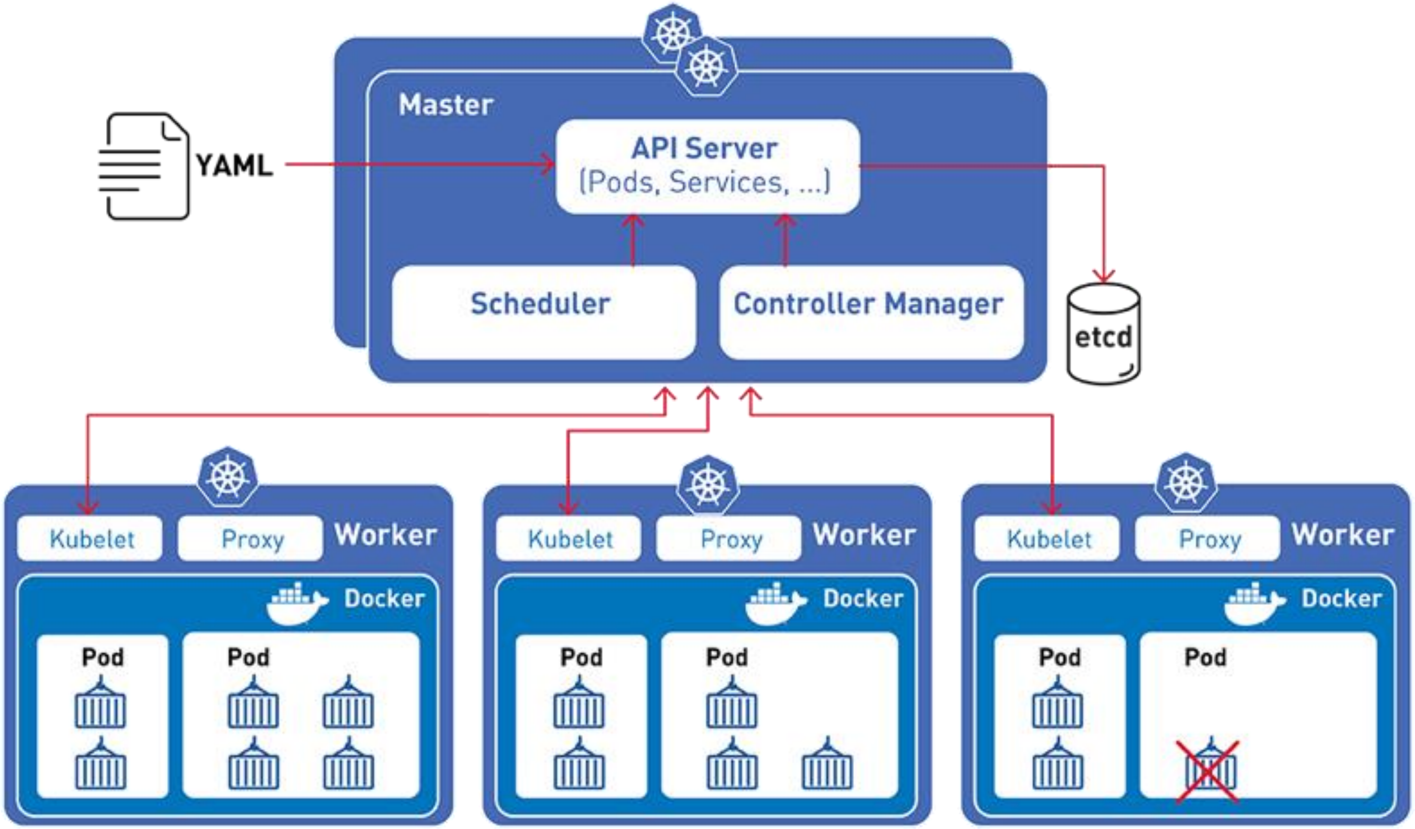
- **Helm**
  - Chart
  - Template
  - Tiller

KUBERNETES

- **Kubernetes (K8s) is an open-source system for:**
  - **automating deployment,**
  - **scaling,**
  - **and management of containerised applications**

Read more: https://kubernetes.io/docs/

KUBERNETES


Master
YAML
API Server
(Pods, Services, ...)
Scheduler
Controller Manager
etcd
Kubelet
Proxy
Worker
Docker
Pod
Pod
Kubelet
Proxy
Worker
Docker
Pod
Pod
Kubelet
Proxy
Worker
Docker
Pod
Pod

KUBERNETES: CONTAINER

- **a Container is a running Docker[1] image in a Kubernetes cluster**

Read more: https://kubernetes.io/docs/concepts/containers/images/

[1] https://www.docker.com/

KUBERNETES: POD

- a Pod is the smallest and simplest unit in the Kubernetes object model that you create or deploy
- a Pod logically represents one machine and contains one or (rarely) more containers

Read more: https://kubernetes.io/docs/concepts/workloads/pods/pod-overview/

KUBERNETES: DEPLOYMENT

- **a Deployment manages a Pod lifecycle and ensures that the required number of Pods is running in the Cluster**
- **When pods crash or are pre-empted, Deployments ensure that they are promptly recreated**

Read more:
https://kubernetes.io/docs/concepts/workloads/controllers/deployment/

KUBERNETES: SERVICE

- **Kubernetes Services are a way to expose an application running on a set of Pods as a network service.**
- **There are three types of services:**
  - **ClusterIp**: exposes an application port on an internal IP, visible only within the cluster
  - **NodePort**: exposes an application port on every node in the cluster
  - **LoadBalancer**: exposes an application port using an external Load Balancer (usually managed by the cloud provider)

Read more: https://kubernetes.io/docs/concepts/services-networking/service/

KUBERNETES: INGRESS

- **Ingress is a way to declaratively define how services should be exposed for the Cluster**
- **They bind domain names to services and can be used to provide Load Balancing, SSL-termination, HTTP-authentication and more.**

Read more: https://kubernetes.io/docs/concepts/services-networking/ingress/

KUBERNETES: PersistentVolumes

- **since files inside Containers are ephemeral, we need a way to provide some persistency to the application**
- **in Kubernetes this is achieved by mounting PersistentVolumes inside a Container.**

Read more: https://kubernetes.io/docs/concepts/storage/volumes/

HELM

- multiple Pods are used in each scenario
- therefore, Helm is used to orchestrate setting up a specific training environment
- this also allows to simplify exercise rollup and clean-up

Read more: https://helm.sh

HELM: CHART

- Helm Charts are used to define how a particular set of Pods will be orchestrated
- a Chart is a set of Kubernetes templates that can be applied and managed using Helm
- it makes development easier and faster since this approach makes it possible to reuse other charts

Read more: https://helm.sh/docs/topics/chart_template_guide/

HELM: TEMPLATE

- Helm Templates are a way to describe a Kubernetes resource in a generic way
- Templates can be parameterized by means of a values.yaml file
- Templates simplifies exercise reuse by encapsulating parts of the deployment into manageable chunks

Read more: https://helm.sh/docs/topics/chart_template_guide/

HELM: TILLER

- **a Tiller is a Pod installed on a Kubernetes Cluster designated to communicate with a K8s cluster on behalf of Helm**

MicroK8s

- **for the exercises, we use a very simple single-node Kubernetes cluster**
- **to simplify the process of VM-creation, we decided to use MicroK8s for cluster deployment**
- **it was initially designed for allowing developers to create their own local environment for testing**
- **but it turned out to be a good choice for the complete exercise setup**

Read more: https://microk8s.io/docs/

THANK YOU FOR YOUR ATTENTION

European Union Agency for Cybersecurity
Vasilissis Sofias Str 1, Maroussi 151 24
Attiki, Greece

☎ +30 28 14 40 9711

✉ csirt-relations@enisa.europa.eu

🌐 www.enisa.europa.eu