



EUROPEAN UNION AGENCY
FOR CYBERSECURITY



ORCHESTRATION OF CSIRT TOOLS

STUDENT TOOLSET – ADMIN MODULES

DECEMBER 2019

ABOUT ENISA

The mission of the European Union Agency for Cybersecurity (ENISA) is to achieve a high common level of cybersecurity across the Union, by actively supporting Member States, Union institutions, bodies, offices and agencies in improving cybersecurity. We contribute to policy development and implementation, support capacity building and preparedness, facilitate operational cooperation at Union level, enhance the trustworthiness of ICT products, services and processes by rolling out cybersecurity certification schemes, enable knowledge sharing, research, innovation and awareness building, whilst developing cross-border communities. Our goal is to strengthen trust in the connected economy, boost resilience of the Union's infrastructure and services and keep our society cyber secure. More information about ENISA and its work can be found at www.enisa.europa.eu.

CONTACT

For contacting the authors, please use csirt-relations@enisa.europa.eu.
PGP Key ID: 31E777EC 66B6052A PGP
For media enquiries about this paper please use press@enisa.europa.eu.

AUTHORS

NASK and Christian Van Heurck (ENISA)

ACKNOWLEDGEMENTS

Hubert Barc (NASK), Jarosław Jedynek (NASK), Paweł Pawliński (NASK), Dominik Sabat (NASK), Krzysztof Stopczyński (NASK) and Iwona Jarosz (NASK).

LEGAL NOTICE

Notice must be taken that this publication represents the views and interpretations of ENISA, unless stated otherwise. This publication should not be construed to be a legal action of ENISA or the ENISA bodies unless adopted pursuant to the Regulation (EU) No 2019/881.

This publication does not necessarily represent state-of-the-art and ENISA may update it from time to time.

Third-party sources are quoted as appropriate. ENISA is not responsible for the content of the external sources including external websites referenced in this publication.

This publication is intended for information purposes only. It must be accessible free of charge. Neither ENISA nor any person acting on its behalf is responsible for the use that might be made of the information contained in this publication.

COPYRIGHT NOTICE

© European Union Agency for Cybersecurity (ENISA), 2020
All material is available under the [Creative Commons BY-NC-SA 4.0 license](https://creativecommons.org/licenses/by-nc-sa/4.0/)¹.

For any use or reproduction of photos or other material that is not under the ENISA copyright, permission must be sought directly from the copyright holders.

¹ <https://creativecommons.org/licenses/by-nc-sa/4.0/>

TABLE OF CONTENTS

1. GDPR GUIDELINES	5
1.1 GENERAL OBSERVATIONS	5
1.2 ADDITIONAL REMARKS ON MISP	6
2. ADMIN PART - GENERAL INFORMATION	7
2.1 INTRODUCTION	7
2.2 CREDENTIALS	7
3. MISP ADMIN	8
3.1 INTRODUCTION	8
3.2 PRECONFIGURED STATES	9
3.2.1 misp-bare	9
3.2.2 misp-configured	9
3.3 EXERCISE	9
3.3.1 Preparation	9
3.3.2 Events	11
3.3.3 Galaxies	12
3.3.4 Taxonomies	12
3.3.5 Adding users	13
3.3.6 Organisation	13
3.3.7 Role permissions	15
3.3.8 Dashboard and Statistics	15
3.3.9 Automation API	15
3.3.10 Additional configuration	16
3.3.11 Synchronisation	16
4. ELASTICSEARCH ADMIN	19
4.1 INTRODUCTION	19
4.2 PRECONFIGURED STATES	19
4.2.1 elasticsearch-bare	19
4.3 EXERCISE: ELASTICSEARCH BASIC ADMINISTRATION	19
4.3.1 Overview of Elasticsearch	19
4.3.2 Overview of Kibana	21
4.3.3 Configure the exercise	22
4.4 GET FAMILIAR WITH ELASTICSEARCH	24



4.4.1	Create an index	24
4.4.2	Adding data to the cluster	27
4.4.3	Health monitoring	30
4.4.4	Bulk insert more test data	31
4.4.5	Exercise: find interesting data in the cluster.	31
4.5	GET FAMILIAR WITH KIBANA	31
4.5.1	Configure index for dashboards	31
4.5.2	Use Kibana to discover your data.	33
4.5.3	Exercise: find interesting data in the cluster.	35
4.5.4	Create a visualisation	35
4.5.5	Exercise: create your own visualisation	37
4.5.6	Real time visualisations	37
5	INTELMQ ADMIN	38
5.1	INTRODUCTION	38
5.1.1	Pipeline	38
5.1.2	Bots	38
5.2	EXCERCISE 1 - CREATE A SIMPLE PIPELINE THAT FETCHES DATA FROM A THIRD PARTY AND OUTPUTS IT TO A LOCAL FILE	39
5.2.1	Enable the installation of IntelMQ	39
5.2.2	Configure the collector	40
5.2.3	Configure the output	40
5.3	EXCERCISE 2 - TEST THE PIPELINE	41
5.4	EXCERCISE 3 - ADD PARSER AND EXPERT BOTS	42
5.4.1	Adding the Parser	42
5.4.2	Adding an Expert	42
5.4.3	Connecting the Bots	43
5.4.4	Check the Output	43
5.5	EXCERCISE 4 - USE MORE COMPLEX COLLECTOR AND OUTPUT BOTS	43
5.5.1	SNARE/TANNER honeypot	43
5.5.2	Adding a custom bot	44
6	THEHIVE ADMIN	47
6.1	INTRODUCTION:	47
6.2	TASKS:	47
6.2.1	Setup accounts	48
6.2.2	Configure Cortex analysers	49
6.2.3	Configure TheHive - Cortex integration.	50
6.2.4	Configure the Hive-MISP integration and check if alerts are fetched	51
6.2.5	Creating a custom Cortex analyser	51
6.2.6	Responders	53
6.2.7	Report templates	53
6.2.8	Case templates	53
6.2.9	Dashboards	54



1. GDPR GUIDELINES

1.1 GENERAL OBSERVATIONS

General observations on GDPR (General Data Protection Regulation) compliance of the tools selected for this training.

Although this topic might seem odd in a technical training at first sight, we wanted to include this because once systems are setup and running, they will potentially be processing large amounts of data, sharing this data with peers inside and outside of your organisation, storing this data in backups and logs.

Adding GDPR compliance and/or concerns as an afterthought will generally lead to less than optimal situations so giving this topic serious considerations before you start implementing systems and procedures is recommended.

The following applies to the training itself but it can easily be transposed to your own situation.

- **Usage of every type of tool selected for this training, involves processing of personal data. The participants should consider themselves as a controller in the meaning of Article 4 point (7) of the GDPR⁴.**

Personally Identifiable Information (PII) types differ depending on the type of tool that is used. However in every case IP addresses, domains, URLs and emails are likely to be processed. Additional details of MISP are provided in section (1.2) below.

Special categories of personal data (in the meaning of Article 9 of the GDPR, *i.e.* revealing racial or ethnic origin, political opinions, religious or philosophical beliefs) are not processed while using the tools selected for this training.

- The processing of PII while using the tools selected for this training is mostly intended for widely understood computer network defence practices. Additional details for MISP are provided in section (1.2) below.
- Unless using the tools selected for this training is not a consequence of a legal obligation to which the participant is subject (Article 6 point (c) of GDPR, *i.e.* national CSIRT's role), the lawfulness of the processing of PII while using these tools could be justifiable as necessary for the purposes of the legitimate interests pursued by the participant or by a third party. As specifically laid down in Recital 49 of GDPR:
The processing of personal data to the extent strictly necessary and proportionate for the purposes of ensuring network and information security, i.e. the ability of a network or an information system to resist, at a given level of confidence, accidental events or unlawful or malicious actions that compromise the availability, authenticity, integrity and confidentiality of stored or transmitted personal data, and the security of the related services offered by, or accessible via, those networks and systems, by public authorities, by computer emergency response teams (CERTs), computer security incident response teams (CSIRTs), by providers of electronic communications networks and services and by providers of security technologies and services, constitutes a legitimate interest of the data controller concerned.

⁴ <https://eur-lex.europa.eu/eli/reg/2016/679/oj>

Having in mind the purposes of processing of PII by the participants while using the tools selected for this training, the legitimate interest of the participants (or a third party) described above should not be found overridden by the interests or fundamental rights and freedoms of the data subjects.

- In general, the participants should not keep the PII that is being processed while using the tools selected for this training in a form which permits identification of data subjects for longer than is necessary for the purposes for which the personal data are processed. Therefore, whenever the purposes of processing of PII by the participant become obsolete (*i.e.* loss of particular legal status), the participant should delete the PII processed while using the tools selected for this training.
- The tools selected for this training do not natively support anonymisation or pseudonymisation so complete deletion of PII processed while using the tools selected for this training is necessary if the purposes of processing of PII by the participant become obsolete.
- The exercise of the rights of data subjects by the controller (the participant) is always dependent on the possibility of identifying by the participant with certainty the person (data subject) who demands exercising its rights on the grounds of GDPR. Bearing in mind that one should not expect an offender to identify themselves before the participant, a possible case of the obligation of reaction to a demand of a data subject is when the victim of a threat is looking for help from the participant. For this event any participant acting as a personal data collector should have prepared a general privacy policy. This is of course less applicable for students-participants but more valid in a professional context.
- In every case participants should verify whether their member state law (or European Union law for that matter) somehow restricts their obligations and rights towards the data subject (as described in Article 23 or Article 89 of the GDPR). If so, some of the above conclusions may not be applicable to this participant in its entirety.

1.2 ADDITIONAL REMARKS ON MISP

While MISP is software dedicated in the first place for information sharing, both PII types processed while using this tool and the purpose of this processing are wider than for the other tools selected for this training.

More details on this can be found in the following publication: "*Information sharing and cooperation enabled by GDPR*".

https://www.misp-project.org/compliance/gdpr/information_sharing_and_cooperation_gdpr.html

For the purpose of this training, it should be noted that except for IP addresses, domains, URLs and emails, other PII may be processed by the participant while using MISP (phone numbers, bank account numbers, etc.). At the same time, the following is stressed by the MISP providers: *In MISP, event attributes are not linked to each other and usually do not enable the identification a data subject by themselves, without additional information. For example, having only an IP address, is usually not enough to identify a data subject without additional information from the ISP. As such, most of the event attributes can be considered as pseudonymised.*

An additional purpose of processing PII by the participant while using MISP, is sharing information between the entities involved with cybersecurity. According to the MISP providers, this activity is also grounded by the Recital 49 of GDPR:

The GDPR actually enables information exchange of personal data between CSIRTs as long as it is consistent with its purposes.

2. ADMIN PART - GENERAL INFORMATION

2.1 INTRODUCTION

This part covers the training modules aimed towards the staff that will be setting up, configuring and maintaining the tools in the portfolio of this training set.

The minimum specifications for a computer that will be used to run the training environment are:

- A 64bit CPU with virtualization support enabled,
- At least 12 GB of RAM,
- Installed a recent version of VirtualBox¹⁵ in the main operating system of the computer,
- 40 GB of free disk space (SSD recommended)..

2.2 CREDENTIALS

The following table gives an overview of the credentials that are needed to access the different systems and tools in the exercises.

Exercise	System	URL	Username	Password
All	Training VM	-	enisa	enisa
MISP admin	MISP1	https://misp.enisa.ex	admin@admin.test	admin
MISP admin	MISP2	https://misp2.enisa.ex	admin@admin.test	Str0ngP@sswd!
MISP analyst	MISP1	https://misp.enisa.ex	admin@admin.test	FirstInstancePassword!
MISP analyst	MISP2	https://misp2.enisa.ex	admin@admin.test	SecondInstancePassword123!
Elasticsearch	Elasticsearch	http://elasticsearch.enisa.ex	-	-
Elasticsearch	Kibana	http://kibana.enisa.ex	-	-
TheHive	TheHive	http://thehive.enisa.ex	admin	admin
TheHive	Cortex	http://thehive.enisa.ex	admin.enisa.ex	admin
TheHive	Cortex	http://cortex.enisa.ex	admin	admin
IntelMQ	IntelMQ	http://intelmq.enisa.ex	-	-
IntelMQ	Honeypot	http://honeypot.enisa.ex	-	-

¹⁵ Oracle VirtualBox virtualisation software can be downloaded for free from this website: <https://www.virtualbox.org/>

3. MISP ADMIN

3.1 INTRODUCTION

Parameter	Description	Duration
Main Objective	Introducing trainees to basic MISP administration concepts. It is targeted at MISP novices. The concepts that will be described include configuring organisation, galaxies, taxonomies, synchronisation and more.	-
Targeted Audience	The exercise is dedicated to members of SOC/CERT/CSIRT teams but also to staff responsible for deployment and maintenance of the platforms.	
Total Duration	2 hours	120 minutes
Time Schedule	Introduction to the exercise	10 minutes
	Basic configuration	15 minutes
	Events	15 minutes
	Galaxies	10 minutes
	Taxonomies	10 minutes
	Roles, Organisations and Synchronisation	60 minutes

This module will introduce you to MISP¹⁶ – a platform for collecting and exchanging IoCs (Indicators of Compromise) and threat information with other organizations.

You will learn about basic concepts related to the tool, such as:

- events,
- attributes,
- objects,
- tags,
- galaxies,
- modules.

Then, you will put that knowledge into practice. You are also going to get familiar with the basic configuration of a MISP instance, including user management and the synchronisation between MISP instances.

This exercise is designed for staff involved in system administration duties, willing to expand their knowledge of MISP internals and basic MISP configuration. It is by no means intended as a full MISP training and it does not cover the installation process of a MISP instance.

For more information on how to install MISP and a complete set of documentation for MISP, we refer to the MISP documentation website¹⁷.

¹⁶ <https://www.misp-project.org/>

¹⁷ <https://www.misp-project.org/documentation/>

3.2 PRECONFIGURED STATES

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

3.2.1 misp-bare

This state consists of two MISP systems.

- The first one is MISP1
- It is reachable at <https://misp.enisa.ex> if you use a browser in your VM environment
- It is not configured at all.
- This represents the bare state of MISP just after installation.
- There is no data in place.
- One account is available with username: `admin@admin.test` and password: `admin`

- The second instance is MISP2
- It is reachable at <https://misp2.enisa.ex>
- It contains data and has a minimal configuration.
- You can login with username: `admin@admin.test` and password: `SecondInstancePassword123!`
- It has the following API Key `gxPEOFh04jGZriMUhBI3U9IyOp7IrxKYifIDMMB3`

3.2.2 misp-configured

This state represents both of the above MISP instances but this time in configured condition. The configuration was done by following the steps hereafter.

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

3.3 EXERCISE

3.3.1 Preparation

Now we will prepare the exercise environment on the Virtual Machine (VM). To start the exercise, first import the virtual machine image using VirtualBox¹⁸ and boot it up. The credentials for the VM are `enisa:enisa`.

3.3.1.1 Reset the state of the exercise

First, we need to reset the state of the exercise by means of a script. Use the terminal in the VM to navigate to the following location:

```
/opt/enisa/trainings-2019/admin/misp
```

Run the following scripts:

```
cd /opt/enisa/trainings-2019/admin/misp
```

followed by;

```
./reset_data.sh.
```

3.3.1.2 Setup the exercise environment

To enable the exercise that contains the two MISP instances, navigate to the following folder:

```
/opt/enisa/trainings-2019/admin/misp
```

¹⁸ <https://www.virtualbox.org/>

Then run the following script:

```
./start-exercise.sh.
```

The environment is ready when the prompt returns, it can take a while for the exercise to start, depending on your virtual machine processing power.

3.3.1.3 Resetting your progress

If needed you can use the following steps to reset any progress you made during the exercise. It is important to **stop** the exercise by issuing the following command:

```
helm delete <id>
```

Where **id** is the chart id that can be obtained with the following command:

```
helm ls.
```

After that do a reset of the progress you made by executing the following script:

```
reset-data.sh
```

3.3.1.4 Basic configuration of MISP

Log into your organization's MISP1 with from within the VM by pointing a web browser to the following URL:

```
https://misp.enisa.ex
```

We start with configuring MISP1 by setting a few simple options. After logging into MISP1, change the default password from:

```
admin@admin.test:admin to Str0ngP@sswd!
```

When the password is changed, you should see confirmation message in MISP on the red background. Please note that something displayed with a red colour in MISP does not always indicate an error - sometimes it is just a message for the user.

Next, we need to set the **baseurl** option (what is configured here will be prepended to all MISP URLs). Several features depend on this being correctly set or else they might not function as expected.

Navigate in MISP1 to the following Settings by using the MISP native menu structure:

- Administration -> Server Settings & Maintenance -> MISP settings -> MISP.baseurl
- There change `https://localhost` to `https://misp.enisa.ex`

Next, we are going to edit the existing **default organisation** parameter so that it has a meaningful name. This is important because the value of this parameter will be displayed all over the place.

- Administration -> List Organisations
- Click on the Edit icon on the far right in the ORGNAME organisation row.
- Edit the name and change it into **MY-SUPER-CERT**
- This value will identify your organisation.
- If you wish, you can fill a brief description of the organisation and complete the optional fields at the bottom of the page.
- Click Submit to save your entries.

After setting the above values and refreshing the site, you can observe an improved state of the system.



When the above configuration is done, you can set the **live** option to **true**, thus enabling non-admin users to access the system.

In a real-world situation, one would wait until everything is configured and verified of course!

In the context of this exercise, we will do it now and as follows;

- Administration -> Server Settings & Maintenance -> MISP settings -> live
- Change **false** to **true**

3.3.2 Events

Events are the core of any MISP instance. They allow you to manage, share and enrich your own intelligence data and that of other organisations.

3.3.2.1 Adding events

To begin, we need to create a **new event**. To do so, we click the **Add Event** option when on the Events list view:

- Event Actions -> Add Event

Here a short description of some of the parameters associated with creating an event

- **Distribution:** defines how far in the chain of synchronized MISP instances the event is going to be published. In practice, this can be defined as the number of hops that the event is going to make before not being distributed further.
 - **This organisation only** (0 hops): only for the organisation of the user who is adding the event.
 - **This community only** (1 hop): all organisations inside the current MISP instance gets the event.
 - **Connected communities** (2 hops): every organisation that is integrated with one of our synchronized organisations.
 - **All communities** (infinite hops): any organisation in the chain of connected organisations.
- **Analysis:** defines if the event is in ongoing analysis or if its analysis has already been completed.
- **Threat Level:** defines level of "importance" of the event. To be interpreted as only a hint for the partition; the exact meaning can vary from organisation to organisation.
 - **Undefined:** No risk
 - **Low:** Mass malware
 - **Medium:** APT malware
 - **High:** Sophisticated APT malware or 0-day attack
- **Event info:** description of the event, ideally with concise info of what happened and/or what the event is about. This is important as this can help other analysts to improve their understanding of the exact details of the event. On the other hand, we want it to be concise so it is easily readable by others.
- **Extends event:** MISP allows for correlation of events so in this field you can put **UUIDs** of other events that correlate to this incident.

After creating an event, we are redirected to the details view. Here we can add **tags, attributes, related events, correlations** and so on.

Attributes are a very important part of an event; they contain information such as *Indicators of Compromise (IoCs)*, *Command & Control Server (C&C) addresses*, *md5 hashes*, or other additional information. Many types of attributes exist. We will focus on events more in the security analyst part (0) of this exercise.

Try to create your own event of choice; it can be anything from a malware campaign to a simple daily report about port scanning.

After completing the fields appropriately, click **Add** to add the event.

When you make appropriate changes to the event and you consider it finished, you can share it with other organisations by clicking on **Publish event** on the left panel.

Now let us see how the event you created presents itself on the events list:

- Event Actions -> List Events

3.3.3 Galaxies

In the next step, we will update the galaxies definition. 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 the MISP galaxy but they can be overwritten, replaced or updated.

3.3.3.1 Enable galaxies

To enable galaxies, follow these steps:

- Galaxies -> Update Galaxies
- Wait for galaxies to update and keep in mind that **this can take a while to complete!**

NOTE: Updating galaxies is only possible with internet access for the VM. This is because updates are performed through a GitHub repository.

After updating the galaxies definitions, we are able to add galaxies to events as follows:

- Go to the detailed event view of the event you created in the previous chapter.
- Scroll down to Galaxies and click on **Add**.
- If the event you created earlier is e.g. related to some banking malware, choose **All namespaces**
- In Select an Option, choose **Banker**, then the appropriate malware family.
- Finally click **Submit**.

You can explore by yourself the available galaxies to find one that is appropriate for the event you created.

3.3.3.2 Examples of galaxies

Here we list some examples of potentially interesting galaxies:

- **Ransomware:** galaxy with information on ransomware campaigns and families, based on the following list that is compiled by security researchers on a voluntary basis:
<https://goo.gl/6e3wia>
- **Threat actor:** characteristics of malicious actors and/or adversaries.
- **Exploit-kit:** list of some well-known exploit kits used by threat actors. The list includes document, browser and router exploit kits. It is not meant to be exhaustive but aims to cover the most seen exploit-kit based threats in the past 5 years.

3.3.4 Taxonomies

A 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. In MISP, there are several taxonomies ready to use, but users can also create their own ones.

3.3.4.1 Enable taxonomies

To enable default taxonomies, click on:

- Event Actions -> List Taxonomies -> Update Taxonomies

NOTE: Updating taxonomies is only possible with internet access for the VM.

After default taxonomies are downloaded from free and open sources, we need to **enable them** in our MISP instance. For the sake of this exercise, we are going to **enable all tags** from one namespace.

To do so find the *stealth_malware* namespace on the list and click on the **plus sign** on the **far right**. This enables the namespace but does not enable the tags inside the namespace. Then click (enable all) on the Active tags column. Now all tags from namespace *stealth_malware* are available to use in the detailed event view.

More information about the tags inside the namespace can be found by clicking on the taxonomy. Click on it and read about the tags meaning.

As with galaxies, we can try them out in our event we created earlier.

Find your event in List Events view once again.

In the tags field click on the plus sign then choose Taxonomy Library: *stealth_malware* and from the field below choose *stealth_malware:type="II"*.

That is the basic use of taxonomies.

Look at the List Events view to see your event now with more information available.

3.3.4.2 Popular taxonomies

- **TLP (Traffic Light Protocol):** classification of sensitive information distribution. There are 4 TLP levels¹⁹:
 - **TLP: RED** personal for named recipients only,
 - **TLP: AMBER** limited distribution,
 - **TLP: GREEN** distributed for particular community,
 - **TLP: WHITE** for unlimited distribution.
- **osint:** Open Source Intelligence - Classification (MISP taxonomies)
- **malware_classification:** classification based on different categories. It is in line with this posting: <https://www.sans.org/reading-room/whitepapers/incident/malware-101-viruses-32848>

3.3.5 Adding users

To add a 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 the user belongs to.
- **Role:** this determines what the user can do in the MISP instance. Read the next section for a quick overview of the MISP permission system.
- Click Submit

3.3.6 Organisation

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

3.3.6.1 Adding a new organisation

To add a new organisation, do the following:

¹⁹ <https://www.first.org/tlp/>



- Click on the Add Organisation button in the administration menu to the left
- Fill out the following fields in the view that is loaded:

New Organisation

If the organisation should have access to this instance, make sure that the Local organisation setting is checked.
If you would only like to add a known external organisation for inclusion in sharing groups, uncheck the Local organisation setting.

Local organisation

Mandatory fields.

Organisation Identifier

Brief organisation identifier No image uploaded for this identifier

Uuid

Paste UUID or click generate Generate UUID

A brief description of the organisation

A description of the organisation that is purely informational.

The following fields are all optional.

Nationality Sector
Not specified For example "financial".

Type of organisation
Freetext description of the org.

Contacts
You can add some contact details for the organisation here, if applicable.

Submit

- **Local organisation:** If the organisation should have full access to this instance, tick the checkbox. If you would only like to add a known external organisation for inclusion in sharing groups, uncheck it.
- **Organisation Identifier:** Name your organisation. If you want to add a picture, you should add a file on the webserver using the 'Server Settings menu'. The picture should have the same name. To learn more about the server settings menu, click [here](#).
- **UUID:** Unique identifier. If you want to share the organisation between MISP multi-instances, use the same UUID.
- **A brief description of the organisation:** Self-explanatory.
- **Nationality:** A drop-down list for selecting the country the organisation belongs to.
- **Sector:** Define the sector of the organisation (Financial, Transport, Telecom...)
- **Type of organisation:** Define the type of the organisation.
- **Contacts:** You can add some contact details for the organisation.

3.3.6.2 Listing all organisations

To list all current organisations of the system, just click on *List Organisations* under the administration menu to the left. There are 3 tabs in this view to filter *local organisations*, *remote organisations* or *both*. The default view displays *local organisations*.

3.3.7 Role permissions

MISP user roles can be found under *Global Actions* -> *Role Permissions* – at this moment all we need is just an admin account.

The **Role Permission** system in MISP consists of following permissions:

- **Site Admin:** Unrestricted access to any data and functionality on this instance.
- **Org Admin:** Limited organisation admin – create and manage users belonging to their own organisation
- **Sync Actions:** Synchronisation permissions can be used to connect two MISP instances and create data on behalf of other users. Make sure that the role with this permission has also access to tagging and tag editing rights.
- **Audit Actions:** Access to the audit logs of the user's organisation.
- **Auth Key Access:** Users with this permission have access to authenticating via their Auth Keys, granting them access to the API.
- **Regex Actions:** Users with this role can modify the regex rules affecting how data is fed into MISP. **Caution is strongly advised with handing out roles that include this permission! User controlled executed regexes are dangerous.**
- **Tagger:** Users with roles that include this permission can attach or detach existing tags to and from events and/or attributes.
- **Tag Editor:** This permission gives users the ability to create, modify or remove tags.
- **Template Editor:** Create or modify templates, to be used when populating events.
- **Sharing Group Editor:** Permission to create or modify sharing groups.
- **Delegations Access:** Allow users to create delegation requests for their own “*org only events*” to trusted third parties.
- **Sighting Creator:** Permits the user to push feedback on attributes into MISP by providing sightings.
- **Object Template Editor:** Create or modify MISP Object templates
- **ZMQ Publisher:** Allow users to publish data to the *ZMQ pubsub* channel via the *publish event to ZMQ* button.

There are predefined roles that you can use when defining users and structure of your organisation, these include:

- **Admin**
- **Org Admin**
- **User**
- **Publisher**
- **Sync user**
- **Read Only**

3.3.8 Dashboard and Statistics

Other system status views are **Dashboard** (*Global Actions* -> *Dashboard*) and **Statistics** (*Global Actions* -> *Statistics*).

So far, these views are empty because there is no data in our organization MISP. However, later they can be used to show system statistics and numbers related to added events and attributes.

3.3.9 Automation API

Automation options can be found in the *Event Actions* -> *Automation* tab. Automation allows for automating tasks using the MISP API.

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

You can read up on this topic on <https://www.circl.lu/doc/misp/automation/#automation-api>.

3.3.10 Additional configuration

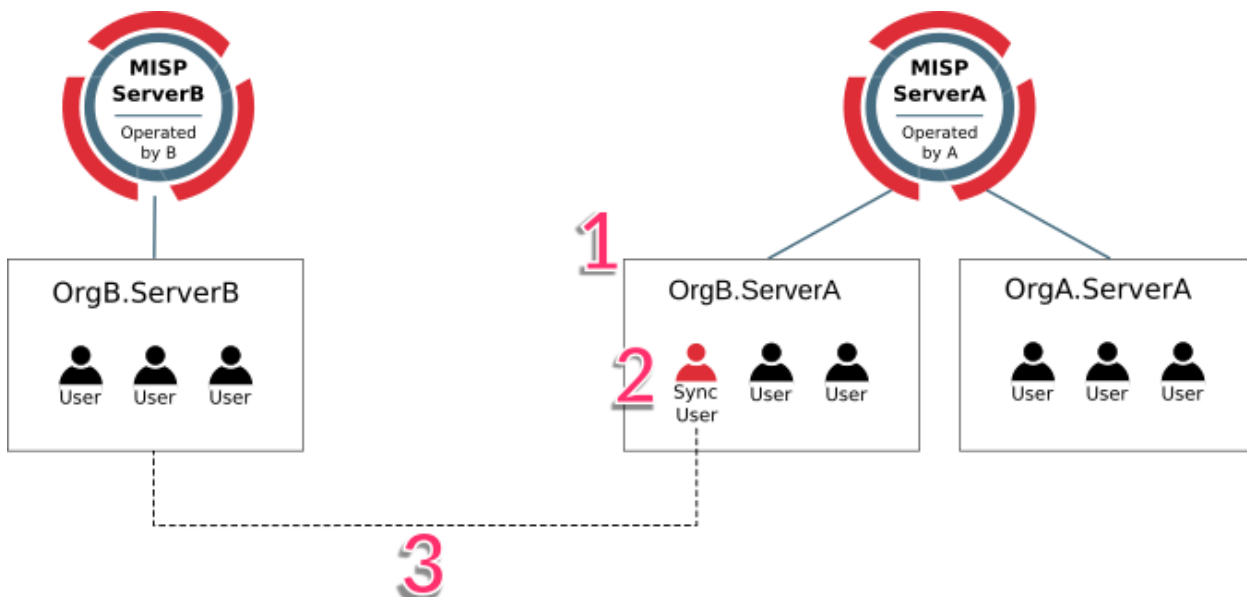
Other MISP settings can be changed later (e.g. plugin options, Redis server configuration for caching, etc...). More information on these settings and configurations can be found in the official MISP documentation²⁰.

Additional system information can be found in *Audit* -> *List Logs*. There may not be many logs now but your actions and the actions of the person preparing the exercise are available there.

3.3.11 Synchronisation

Synchronisation allows exchanging data between MISP instances. This can improve cooperation between organisations and allow for a smooth and fast IoC and/or data exchange. On the following image, you see the concept of a synchronisation setup.

Figure 1: Synchronisation setup



Legend:

- Synchronisation between two MISP servers
- Organisation in the MISP database of a MISP server
- Organisation
- 👤 User of an organisation in the MISP database of a MISP server
- 🌀 MISP server (also called MISP instance)

The common way for synchronizing MISP instances is as follows:

- **Step 1:** Add OrgB as a local organisation on ServerA (OrgB.ServerA) using OrgB's existing UUID from their local organisation on ServerB.
- **Step 2:** Add a Sync User (syncuser@OrgB.ServerA) in the organisation OrgB.ServerA on the MISP ServerA.
- **Step 3:** Set up a sync server on MISP ServerB using the key (called Authkey) from the sync user (syncuser@OrgB.ServerA) created on MISP ServerA.

²⁰ <https://www.misp-project.org/documentation/>

We will configure our MISP instance to perform automatic synchronisation with a remote instance. For that purpose, we will use the second MISP instance (MISP2) that is accessible via your VMs browser at the following URL:

- <https://misp2.enisa.ex>.
- Credentials are: *admin@admin.test* and *SecondInstancePassword123!*

First, we need to create the *Local organisation* representing the organisation we want to synchronise from, as explained in **Step 1** above.

To do this, login to <https://misp2.enisa.ex> and go to *Administration -> Add Organisations*, then we fill in the form with following data.

- Uncheck *Local organisation*.
- Fill in the name *MY-SUPER-CERT*.
- Set the UUID to the UUID of the MISP1 located at <https://misp.enisa.ex>, it should be equal to *5d19ecf9-1e78-49fe-9d31-0091ac110002*. This is **very** important!
- Click *Submit* to create.

Next, we need to create a Sync User on our remote instance, so create a user with the following parameters:

- Email: *sync-user@my-super-cert.ex*.
- Organisation: *MY-SUPER-CERT*.
- Role: *Sync User*.
- Click *Submit*.

Now save the *Authkey* that is generated automatically for the *Sync User*. It should be in the following format *iHRWvgk3aSSPxGatzLbfVYwQkNA48s4vapAwc52P*.

Now move back to MISP1 and do the following steps:

- Go to *Sync Actions -> List Servers -> New Servers*

Then we need to set URL of the other instance we want to access (MISP2).

- Set *Base url* to <https://misp2.enisa.ex> and *Instance name* to *EXTERNAL-PROVIDER-X*.
- Set *Remote Sync Organisation Type* to *Local organisation* and *Owner of remote instance* to *MY-SUPER-CERT*.
- Set *Authkey* to the value obtained while creating the Sync User.
- Check *Push and Pull* in the *Enabled synchronisation methods*. This allows for two-way communication>
Remember that any sharing options that were described earlier apply here as well.
Unpublished events are not going to be visible.
Important: there are multiple ways to setup the synchronisation. The way you choose to do does **NOT** change the Push/Pull behaviour!
- Check *Self Signed* in *Misc settings*. This allows for self-signed MISP certificates. In a real production environment, this can probably be omitted for obvious reasons.
- Click *Submit*.
- Click *Run* under *Connection test*.

You should get an output that is similar to the following:

```
Local version: 2.4.103
Remote version: 2.4.103
Status: OK
Compatibility: Compatible
POST test: Received sent package
```

If this is indeed the case then the synchronisation is set!

Otherwise, check if you followed the above steps correctly and on the right MISP instances (MISP1 and MISP2). You can also check this [GitHub issue for more information](#).

We can now see the effects of the synchronisation:

- Login to <https://misp2.enisa.ex>, choose an event and click *Publish* on the left panel.
- Go back to MISP1 and in the *Sync actions* -> *List Servers* find and press the *Pull all* button. This should pull published event from MISP2. Alternatively, you can wait a bit for the sync to happen automatically.
- You can now observe in MISP1 what has changed after the sync process is completed.

4. ELASTICSEARCH ADMIN

4.1 INTRODUCTION

Parameter	Description	Duration
Main Objective	Elasticsearch administration workshop introduces trainees to basic Elasticsearch administration concepts. It is targeted at Elasticsearch novices. The concepts being described include index creation, health checking and management. In a further part of the exercise, Kibana is introduced as a web frontend for Elasticsearch cluster management and discovery.	-
Targeted Audience	The exercise is dedicated to members of SOC/CERT/CSIRT teams but also to staff responsible for deployment and maintenance of the platforms.	
Total Duration	2 hours	120 minutes
Time Schedule	Introduction	30 minutes
	Elasticsearch: getting started and exercises	30 minutes
	Kibana: getting started	30 minutes
	Kibana: exercises	30 minutes

This exercise is designed for the administrators willing to expand their knowledge about Elasticsearch internals and configuration.

4.2 PRECONFIGURED STATES

4.2.1 elasticsearch-bare

This represents the RAW and un-configured Elasticsearch and Kibana instances:

- Elasticsearch is installed and working, but there is no data inside.
- Kibana is installed and connected to Elasticsearch but no further configuration was done.

4.3 EXERCISE: ELASTICSEARCH BASIC ADMINISTRATION

4.3.1 Overview of Elasticsearch

Elasticsearch²¹ is a distributed search and analytics engine designed for fast full text and structured search. Elasticsearch exposes a REST²² API²³ that can be used directly by CURL²⁴, or it can be accessed with a programming language.

Elasticsearch can be used to store large amounts of structured data while allowing querying this data efficiently. Elasticsearch is not a traditional relational database so to understand how it works you need to familiarise yourself with a few basic Elasticsearch concepts:

- **Document** is the most basic entity in Elasticsearch. It represents a single piece of information that can be indexed and it is roughly comparable to a *row* in a traditional

²¹ <https://www.elastic.co/>

²² REST: Representational state transfer. It's a style of creating HTTP services, especially popular for API-heavy application

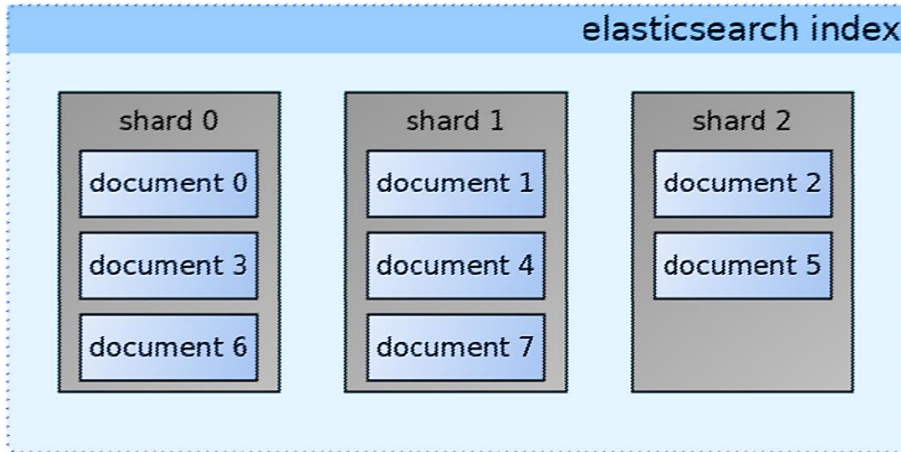
²³ API: Application Programming Interface - interface exposed by the application for the programmers

²⁴ curl: <https://curl.haxx.se/>. Command line tool for sending requests using multiple supported protocols. Most commonly used with HTTP

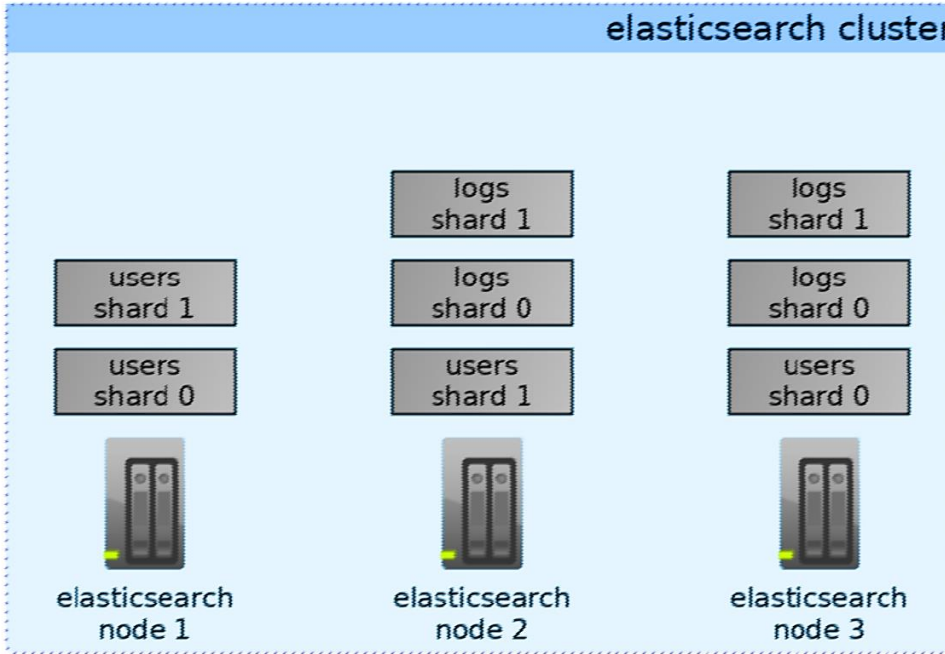
database.

Examples: a single customer, a single blog post a single log entry.

- **Index** is roughly similar to a table in a traditional database. It is used to store a collection of documents with a similar structure. An index is identified by its lowercase name



- **Type** is a *deprecated* concept, mentioned here only to avoid confusion created by obsolete tutorials. Before Elasticsearch 6.x, a single index could contain multiple document types with different schemas (so back then an index used to be more like a database than a table). Beginning with Elasticsearch 6.x, support for this mechanism is limited, and it will be completely removed in the future.
- **Node** is a single Elasticsearch server in a cluster. It can host multiple indexes and is used to physically store index data on the disk. It is identified by a name (by default, a random UUID).
- **Cluster** is a collection of one or more nodes. It can be used to query all your data that is distributed among your nodes. For optimal performance and reliability, it is recommended to have at least *three* nodes in your cluster. However, it is possible to have a cluster with only one node - that is what we will do in this exercise. In general, if you do not have a large amount of data and you do not need high availability, you can start with a single node and scale horizontally to multiple nodes later when the need arises.
- **Shards** are a way to split indexes into smaller pieces. In some cases, indexes can become so huge that it is impractical to store them on a single node. To solve this problem, you can split your index into a predefined number of shards. Each shard could potentially be stored on a different node. When Elasticsearch does a query related to an index, it will check all of its shards in parallel and merge the results. For high availability environments, it is possible to create one or even more replicas for every shard. In this case, every shard will be stored on more than one server, and taking a single node offline will not affect the cluster negatively.



Most of these concepts are also present in traditional relational databases. The following table may be helpful:

Relational Database Concept	Elasticsearch Name
Table	Index
Table Row	Document
Database Server	Node (Elasticsearch Server)
Database Cluster	Elasticsearch Cluster

4.3.2 Overview of Kibana

The Elasticsearch API is not designed to be used by humans, so it is recommended to additionally set up a user-friendly web interface. The most popular tool used for this purpose is Kibana²⁵. Kibana is an open source visualisation platform used in combination with Elasticsearch to browse, search and analyse collected data. We will cover configuration and usage of Kibana later.

In this exercise, we will cover configuration of the tools, and basic administrative tasks.

²⁵ <https://www.elastic.co/products/kibana>

4.3.3 Configure the exercise

4.3.3.1 Ensure that DNS is configured properly.

Subdomains of `.enisa.ex` should have a valid A-record:

```
$ dig -ta +short
elasticsearch.enisa.ex 127.0.0.1
```

```
$ dig -ta +short
kibana.enisa.ex
127.0.0.1
```

4.3.3.2 Apply the helm configuration file

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

4.3.3.3 Wait for the deployment to complete.

Be patient! It can take a few minutes before the tools are downloaded and ready.

4.3.3.4 Ensure that Elasticsearch works correctly.

Either point your browser to <http://elasticsearch.enisa.ex>:

Alternatively, you can use the command line to issue the following curl command:

JSON	Raw Data	Headers
Save	Copy	Collapse All
Expand All	Filter JSON	
name: "KdBMg1m"		
cluster_name: "docker-cluster"		
cluster_uuid: "Aas-B_GVQC2MEh40gM4Xw"		
▼ version:		
number: "6.6.1"		
build_flavor: "default"		
build_type: "tar"		
build_hash: "1fd8f69"		
build_date: "2019-02-13T17:10:04.160291Z"		
build_snapshot: false		
lucene_version: "7.6.0"		
minimum_wire_compatibility_version: "5.6.0"		
minimum_index_compatibility_version: "5.0.0"		
tagline: "You Know, for Search"		

```
$ curl elasticsearch.enisa.ex

{
  "name" : "xkJSyKR",
  "cluster_name" : "docker-cluster", "cluster_uuid" :
  "pQ06hyg0SyuYwb07Rxnwkw",
  "version" : {
    "number" : "6.6.1",
    "build_flavor" : "default",
    "build_type" : "tar",
```



```

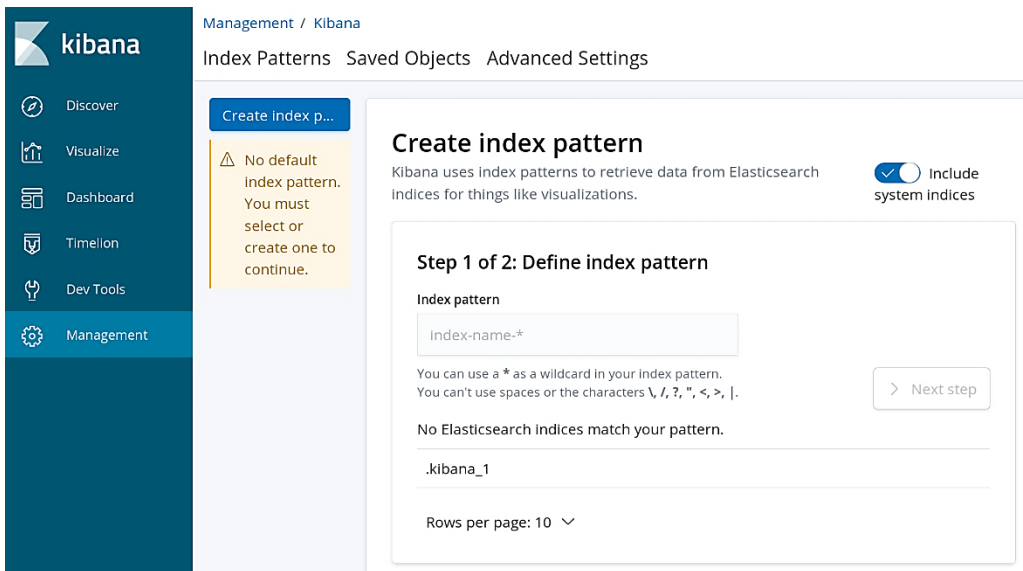
    "build_hash" : "1fd8f69",
    "build_date" : "2019-02-13T17:10:04.160291Z",
    "build_snapshot" : false,
    "lucene_version" : "7.6.0",
    "minimum_wire_compatibility_version" : "5.6.0",
    "minimum_index_compatibility_version" : "5.0.0"
  },
  "tagline" : "You Know, for Search"
}

```

4.3.3.5 Ensure that Kibana works correctly.

Either point your browser to <http://kibana.enisa.ex>:

Alternatively, you can use the command line to issue the following curl command:



```
$ curl kibana.enisa.ex/ -v
```

```
* Connected to kibana.enisa.ex (195.187.123.210) port 80 (#0)
```

```
> GET / HTTP/1.1
```

```
> Host: kibana.enisa.ex
```

```
> User-Agent: curl/7.58.0
```

```
> Accept: */*
```

```
>
```

```
< HTTP/1.1 302 Found
```

```
< Server: nginx/1.15.10
```

```
< Date: Tue, 02 Jul 2019 06:28:35 GMT
```

```
< Content-Type: text/html; charset=utf-8
```

```
< Content-Length: 0
```

```
< Connection: keep-alive
```

```
< location: /app/kibana
```

```
< kbn-name: kibana
```

```
< cache-control: no-cache
```

```
<
```

4.4 GET FAMILIAR WITH ELASTICSEARCH

4.4.1 Create an index

In this exercise, we will work with a simple index simulating parsed access logs from your website. While it is possible to insert data into Elasticsearch without explicitly creating an index beforehand, it is not recommended and it often leads to a bad performance (Elasticsearch tries to create a default index and has to make some guesses on the nature of your data). Therefore, you should always create indexes before inserting data into the cluster.

As an exercise, let us create a simple index for access logs. An access log is a list of important information about all the requests coming to the system that generated it. Web servers and other internet services usually generate it. They can look like this:

```
123.123.123.123 - - [08/Aug/2019:06:54:10 +0000] "GET /blog/my-first-post/
HTTP/1.1" 200 34677

"https://www.google.com/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6)
AppleWebKit/605.1.15 (KHTML, like Gecko) Version/12.1.2 Safari/605.1.15"
"10.244.0.1"

123.123.123.123 - - [08/Aug/2019:06:54:10 +0000] "GET /css/main.css HTTP/1.1" 200 2714
"https://my-blog.net/blog/my-first-post/" "Mozilla/5.0 (Macintosh; Intel Mac OS X
10_14_6) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/12.1.2 Safari/605.1.15"
"10.244.0.1"

123.123.123.123 - - [08/Aug/2019:06:54:11 +0000] "GET /images/favicon.png
HTTP/1.1" 200 8595 "https://my-blog.net

/blog/my-first-post/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6)
AppleWebKit/605.1.15 (KHTML, like Gecko) Version/12.1.2 Safari/605.1.15"
"10.244.0.1"

123.123.123.123 - - [08/Aug/2019:06:56:27 +0000] "HEAD / HTTP/1.1" 200 0
"http://tailcall.net" "Mozilla/5.0+ (compatible; UptimeRobot/2.0;
http://www.uptimerobot.com/)" "10.244.0.1"
```

The above is a very common access log format, used by all major web servers. In this case, we can see the following:

- Four requests to the system, all from the IP 123.123.123.123.
- Everything happened on 2019-08-08, between 6:54:10 and 6:54:10.
- Three HTTP GET requests (download content), and one HEAD request (metadata). All requests returned HTTP 200 status code (i.e. success)
- The first request had the biggest response - 34677 bytes were downloaded and the visitor came to the website from google.com (judging by referrer).
- We can also see the useragent of the visitor, and thus deduce the OS and browser they are using.

To create our index, we will need a mapping for three fields:

- timestamp
- date ip , of type ip
- url , of type text



Elasticsearch exposes a custom HTTP API. It is possible to communicate with it using numerous libraries for various programming languages, or user-friendly tools (notably, Kibana), but for now we are going to use it directly to understand how it works. Another benefit is that the only external tool we will need is curl, which is a standard UNIX²⁶ tool and thus present on almost every machine.

Let us get to work. Please execute the following command in a terminal:

```
curl -X PUT "elasticsearch.enisa.ex/logs" -H 'Content-Type: application/json' -d'
```

```
{
  "settings" : {
    "number_of_shards"
    : 1
  },
  "mappings": {
    "request": {
      "properties": {
        "timestamp":
        {
          "type": "date"
        },
        "ip": {
          "type": "ip"
        },
        "url": {
          "type": "text"
        }
      }
    }
  }
}
```

²⁶ <https://opensource.com/article/18/5/differences-between-linux-and-unix>



As a response, you should get something similar to this:

```
{"acknowledged":true,"shards_acknowledged":true,"index":"logs"}
```

This has created an index called *logs* with a single mapping called *request*. As mentioned earlier, due to deprecation of types (since Elasticsearch 6.x) it is not possible to have multiple mappings in a single index. You can verify that the index exists by using the API to query it.

```
curl -X GET "elasticsearch.enisa.ex/logs" -H 'Content-Type: application/json' -d'
```

```
{
  "query": { }
}
```

The response should be similar to:

```
{"logs":{"aliases":{},"mappings":{"request":{"properties":{"ip":{"type":"ip"},"timestamp":{"type":"date"},"url":{"type":
```

This is not very easy to read. It is possible to change the output format of Elasticsearch commands by adding an optional GET parameter. For example, adding a *?pretty=true* or just *?pretty* will pretty-print the output. Let us try it:

```
curl -X GET "elasticsearch.enisa.ex/logs?pretty" -H 'Content-Type: application/json' -d'
```

```
{
  "query": { }
}
```

The response is much more readable now:

```
{
  "logs" : {
    "aliases" : { },
    "mappings" : {
      "request" : {
        "properties" : {
          "ip" : {
            "type" : "ip"
          },
          "timestamp" : {
```



```
      "type" : "date"
    },
    "url" : {
      "type" : "text"
    }
  }
},
"settings" : {
  "index"
  : {
    "creation_date" : "1565285637845",
    "number_of_shards" : "1",
    "number_of_replicas" : "1",
    "uuid" :
    "PjPrBUX3SL27rqgrioha
    Dw", "version" : {
      "created" : "6060199"
    },
    "provided_name" : "logs"
  }
}
}
```

The query succeeded, but there is no data in the index yet - we have to add it first. We also get some metadata about the index, like a creation date, number of shards & replicas and index schema.

4.4.2 Adding data to the cluster

Let us add some data using the HTTP API directly again. Execute the following bash commands:

```
curl -XPOST http://elasticsearch.enisa.ex/logs/request/1 -H 'Content-
Type: application/json' -d '
{
  "timestamp": "2019-07-
  01T12:10:30Z", "ip":
  "10.0.0.1",
  "url": "/"
```



```
}'  
curl -XPOST http://elasticsearch.enisa.ex/logs/request/2 -H 'Content-  
Type: application/json' -d '  
{  
  "timestamp": "2019-07-  
01T12:10:31Z", "ip":  
  "10.0.0.1",  
  "url": "/favicon.ico"  
}'  
curl -XPOST http://elasticsearch.enisa.ex/logs/request/3 -H 'Content-  
Type: application/json' -d '  
{  
  "timestamp": "2019-07-  
01T12:10:32Z", "ip":  
  "10.0.0.1",  
  "url": "/robots.txt"  
}'  
curl -XPOST http://elasticsearch.enisa.ex/logs/request/4 -H 'Content-  
Type: application/json' -d '  
{  
  "timestamp": "2019-07-  
01T12:10:33Z", "ip":  
  "10.0.0.2",  
  "url": ""  
}'
```

Now let us verify that the inserted data is there. You can easily get data by ID, so let us look at the request with *id 1*.

```
curl -XGET "http://elasticsearch.enisa.ex/logs/request/1?pretty" -H  
'Content-Type: application/json' -d '  
{  
  "_index" : "logs",  
  "_type" : "request",  
  "_id" : "1",  
  "_version" : 1,  
  "_seq_no" : 0,  
  "_primary_term" : 1,  
  "found" :  
  true,  
  "_source" : {
```



```
"timestamp" : 1562065617,  
"ip" : "10.0.0.1",  
"url" : "/"  
}  
}  
,
```

You can also try to do simple queries using the API. The basic query format looks like this:

```
curl -XGET "http://elasticsearch.enisa.ex/logs/_search?pretty" -H  
'Content-Type: application/json' -d '  
{  
  "query" : {  
    "term" : { "ip" : "10.0.0.2" }  
  }  
}'
```

There is also a shortcut form, which is quite useful when querying Elasticsearch manually:

```
curl -XGET  
"http://elasticsearch.enisa.ex/logs/_search?q=ip:10.0.0.2&pretty"
```

Both forms are equivalent and should return something similar to:

```
{  
  "took" : 2,  
  "timed_out"  
  : false,  
  "_shards" :  
  {  
    "total" : 1,  
    "successful" : 1,  
    "skipped" : 0,  
    "failed" : 0  
  },  
  "hits" : {  
    "total" : 1,  
    "max_score" : 1.0,  
    "hits" : [  
      {  
        "_index" : "logs",  
        "_type" : "request",
```




```
"_id" : "4",
  "_score" : 1.0,
  "_source" : {
    "timestamp" : "2019-07-
01T12:10:33Z", "ip" :
    "10.0.0.2",
    "url" : "/"
  }
}
```

4.4.3 Health monitoring

An important part of cluster administration is monitoring. Elasticsearch exposes a handy endpoint that returns all the important information about your cluster:

```
$ curl "elasticsearch.enisa.ex/_cluster/health?local=true" | jq
{
  "cluster_name":
  "docker-cluster",
  "status": "green",
  "timed_out":
  false,
  "number_of_nodes":
  1,
  "number_of_data_nodes": 1,
  "active_primary_shards": 1,
  "active_shards": 1,
  "relocating_shards": 0,
  "initializing_shards": 0,
  "unassigned_shards": 0,
  "delayed_unassigned_shards": 0,
  "number_of_pending_tasks": 0,
  "number_of_in_flight_fetch": 0,
  "task_max_waiting_in_queue_millis": 0,
  "active_shards_percent_as_number": 100
}
```

For a healthy cluster, the status should be "green", number of pending tasks should be small, and the number of unassigned shards should be zero.



In a real world scenario, this should be integrated into a full-blown monitoring solution like Nagios²⁷ or Icinga²⁸. We will not cover the monitoring configuration here, but integrations with most of the industry standard solutions are already freely available on the internet.

4.4.4 Bulk insert more test data

We will need plenty of data for the next exercise. Please execute the `upload.py` script from the `./admin/elasticsearch/exercise/basics` directory.

```
$ python3 upload.py
```

Be patient since this can take a while to complete.

4.4.5 Exercise: find interesting data in the cluster.

Using the Elasticsearch query syntax that we practiced in 4.4, answer to the following questions:

- What was the IP of the bot that tried to download the `/wp-config.bak` file?
 - 195.187.238.213
- How many requests to `/wp-login.php` were performed?
 - 21
- Find all requests performed by the user with IP 195.187.238.221
 - requests to:
 - "url": "/?author=1"
 - "url": "/?author=2"
 - "url": "/?author=3"
 - "url": "/?author=4"
 - "url": "/?author=5"
 - "url": "/?author=6"
 - "url": "/?author=7"
 - "url": "/?author=8"
 - "url": "/?author=9"
 - "url": "/?author=10"
 - "url": "/?author=11"
 - "url": "/?author=12"
 - "url": "/?author=13"
 - "url": "/?author=14"
 - "url": "/?author=15"

4.5 GET FAMILIAR WITH KIBANA

4.5.1 Configure index for dashboards

First, click on a "dashboard" button on the left of the screen.

Kibana uses index patterns to retrieve data from Elasticsearch. Before we start using it, we need to configure a valid index pattern.

Index patterns tell Kibana which indexes we want to use. An index pattern can match a single index, but it can also be a wildcard (which is useful, when we have indexes sharded by month, for example).

²⁷ <https://www.nagios.org/>

²⁸ <https://icinga.com/>



In our case, the only index we are using is *logs*. Let us add it. Type "logs" into an index pattern field, and press the button at the centre of the page:

[Index Patterns](#) [Saved Objects](#) [Advanced Settings](#)

Create index p...

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

Create index pattern

Kibana uses index patterns to retrieve data from Elasticsearch Include system indices

Step 1 of 2: Define index pattern

Index pattern

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

> Next step

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

logs

Rows per page: 10 ▾

Next step is a *time filter field configuration*. Select "timestamp".

[Management / Kibana](#)

[Index Patterns](#) [Saved Objects](#) [Advanced Settings](#)

Create index p...

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

Create index pattern

Kibana uses index patterns to retrieve data from Elasticsearch Include system indices

Step 2 of 2: Configure settings

You've defined **logs** as your Index pattern. Now you can specify some settings before we create it.

Time Filter field name Refresh

 ▾

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.

> Show advanced options

< Back [Create index pattern](#)



This is all we need to do now. The index is properly configured for Kibana now:

★ logs
★ ↻ 🗑️

🕒 Time Filter field name: timestamp

This page lists every field in the **logs** index and the field's associated core type as recorded by Elasticsearch. To change a field type, use the Elasticsearch [Mapping API](#) 📄

Fields (10)

Scripted fields (0)

Source filters (0)

All field types ▾

Name	Type	Format	Searcha...	Aggregat...	Excluded
_id	string		●	●	✎
_index	string		●	●	✎
_score	number				✎
_source	_source				✎
_type	string		●	●	✎
ip	string		●		✎
ip.keyword	string		●	●	✎
timestamp 🕒	date		●	●	✎

4.5.2 Use Kibana to discover your data.

Click a "discover" option on the left. Change the time range in the top right corner, and look at your data.



The screenshot shows the Kibana search interface. At the top, there are 15,126 hits. The search bar contains a query: `>_ Search... (e.g. status:200 AND extension:PHP)`. A filter for "Last 5 years" is applied. The left sidebar shows navigation options: Discover, Visualize, Dashboard, Timelion, Dev Tools, and Management. The main content area displays "logs*" with a "Selected fields" section showing `? _source`. Below this, a bar chart shows the count of logs over time, with a significant spike in early 2019. The x-axis is labeled "timestamp per 30 days" and the y-axis is "Count". Below the chart, a table of results is shown with columns for "Time" and "_source".

Time	_source
▶ August 8th 2019, 17:55:27.000	<code>timestamp: August 8th 2019, 17:55:27.000 ip: 95.203.186.253 url: / _id: 15125 _type: request _index: logs _score: -</code>
▶ August 8th 2019, 17:50:27.000	<code>timestamp: August 8th 2019, 17:50:27.000 ip: 151.156.98.8 url: / _id: 15124 _type: request _index: logs _score: -</code>

In this view, you can use the Lucene²⁹ query syntax, which is noticeably easier than the Elasticsearch DSL³⁰. For example, you can find requests with a specified IP using the following query:

`ip:"173.70.75.44"`

The screenshot shows the Kibana search interface with the query `>_ ip:"173.70.75.44"` entered in the search bar. The left sidebar shows the "logs*" index pattern. The main content area displays a bar chart showing the count of logs for the specified IP over time, with a peak in early 2019. Below the chart, a table of results is shown with columns for "Time" and "_source".

Time	_source
▶ August 8th 2019, 09:18:44.000	<code>timestamp: August 8th 2019, 09:18:44.000 ip: 173.70.75.44 url:</code>
▶ August 8th 2019, 06:54:10.000	<code>timestamp: August 8th 2019, 06:54:10.000 ip: 173.70.75.44 url:</code>
▶ August 8th 2019, 06:16:54.000	<code>timestamp: August 8th 2019, 06:16:54.000 ip: 173.70.75.44 url:</code>

You can also query for a requests from a specific day:

- `timestamp:"2019-08-07"`

Alternatively, even a time range:

²⁹ <https://lucene.apache.org/>

³⁰ https://elasticsearch-dsl.readthedocs.io/en/latest/search_dsl.html

- timestamp:["2019-08-07" TO *]

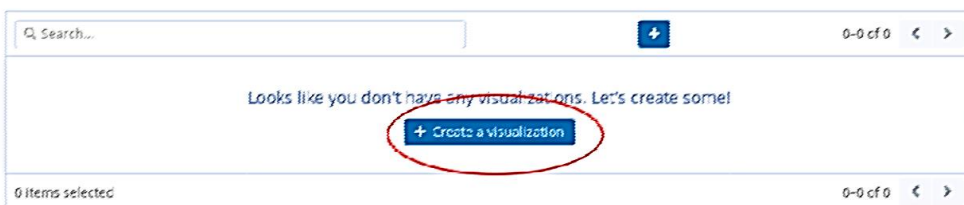
4.5.3 Exercise: find interesting data in the cluster.

Using the Lucene query syntax that we practiced in 4.5.2, answer to the following questions:

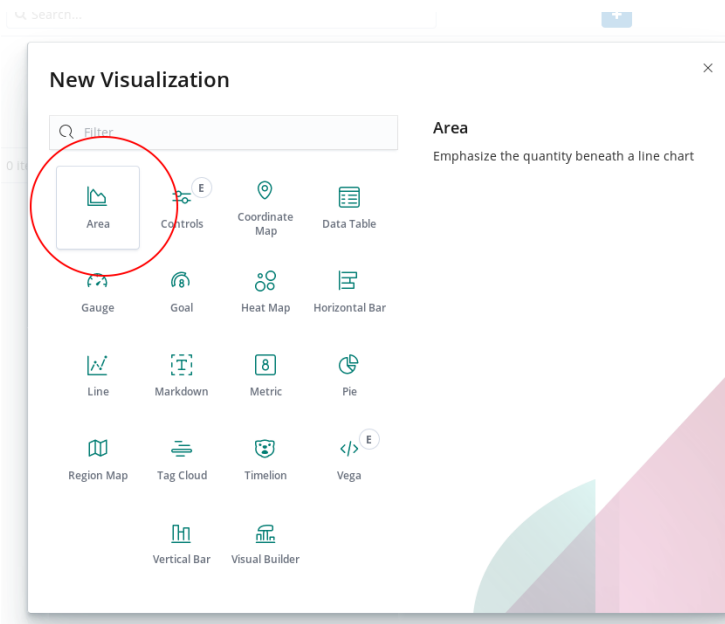
- What was the IP of the bot that tried to download the /wp-config.bak file?
- How many requests to /wp-login.php were performed?
- Find all requests performed by the user with IP 195.187.238.221 .

4.5.4 Create a visualisation

Now let us add a simple visualisation. Click a "visualisation" option on the left, and then the "create a visualisation" button.



Select an area chart:



Configure the x-axis. The most common way to bucket the x-axis is to use a date histogram:



logs*

Data Metrics & Axes Panel Settings ▶ ✕

Metrics

Y-Axis Count

Add metrics

Buckets

X-Axis

Aggregation: Date Histogram help

Date Histogram

Field: timestamp

Interval: Auto

Drop partial buckets

Custom Label

Advanced

Add sub-buckets

Finally, select a good time range to match the data:

Save Share Inspect Refresh Auto-refresh Last 6 months

Time Range

Quick Relative Absolute Recent

From: 2019-07-01 19:54:52.300 To: 2019-08-10 19:54:52.300

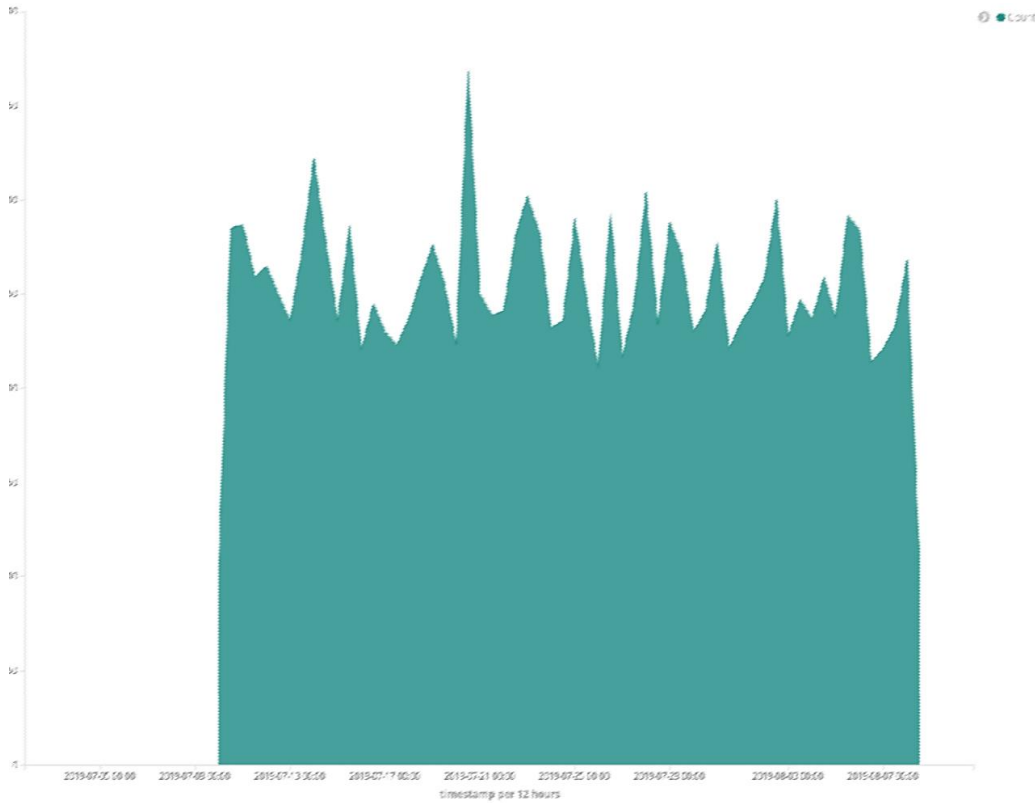
July 2019							August 2019						
Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat
	01	02	03	04	05	06					01	02	03
07	08	09	10	11	12	13	04	05	06	07	08	09	10
14	15	16	17	18	19	20	11	12	13	14	15	16	17
21	22	23	24	25	26	27	18	19	20	21	22	23	24
28	29	30	31				25	26	27	28	29	30	31

Go



The final result should look like the following figure.

Figure 2: Amount of requests done on every day. Spikes on the chart represent spikes in the traffic



4.5.5 Exercise: create your own visualisation

Create a pie chart by grouping requests by IP. One of the IPs should stand out.

- Can you tell which one is it? Hint: you need to select a proper aggregation method and field.
- Remember to increase the number of buckets

4.5.6 Real time visualisations

Kibana has many more powerful features than we have covered during this exercise. Among the most-useful ones are real-time dashboards that allow analysts to spot new incoming events and trends in data in real-time. To enable real-time dashboards, you need to turn on the *Auto Refresh* feature and everything should work automatically.



5. INTELmq ADMIN

5.1 INTRODUCTION

Parameter	Description	Duration
Main Objective	This exercise introduces IntelMQ: platform for automated data processing. Trainees are going to get familiar with IntelMQ, SNARE/TANNER and related concepts.	-
Targeted Audience	The exercise is dedicated to (new) CSIRT staff involved in incident handling.	
Total Duration	1,5 hours	90 minutes
Time Schedule	Introduction	10 minutes
	Task 1: Creating and testing a simple pipeline	20 minutes
	Task 2: Introducing new nodes	30 minutes
	Task 3: More complex pipeline	30 minutes

IntelMQ is a system for incident response team to collect, process and analyse data from various sources (e.g. Indicators of Compromise (IoCs), Command & Control servers (C&C), suspicious IP addresses etc.) using a message queue protocol. Its advantage over similar applications (like Logstash³¹) is that it contains many predefined modules that allow fetching of formatted data from many external sources.

5.1.1 Pipeline

Data processing in IntelMQ is realised by the pipeline mechanism. The input is consumed, processed and presented using advanced and well-suited models for processing unstructured data sets.

In this exercise, we will get familiar with IntelMQ interface by trying to create a complete pipeline.

First, we will gather data from a web application honeypot - SNARE³² in our case.

Next, we will load data generated by SNARE into IntelMQ and process it: parse, de-duplicate and enrich it with additional data like geolocation.

At the end, we will output the results to Elasticsearch for convenient browsing.

5.1.2 Bots

The whole idea of IntelMQ is based on so-called **bot nodes** and the connections between them.

³¹ <https://www.elastic.co/products/logstash>

³² <https://github.com/mushorg/snare>

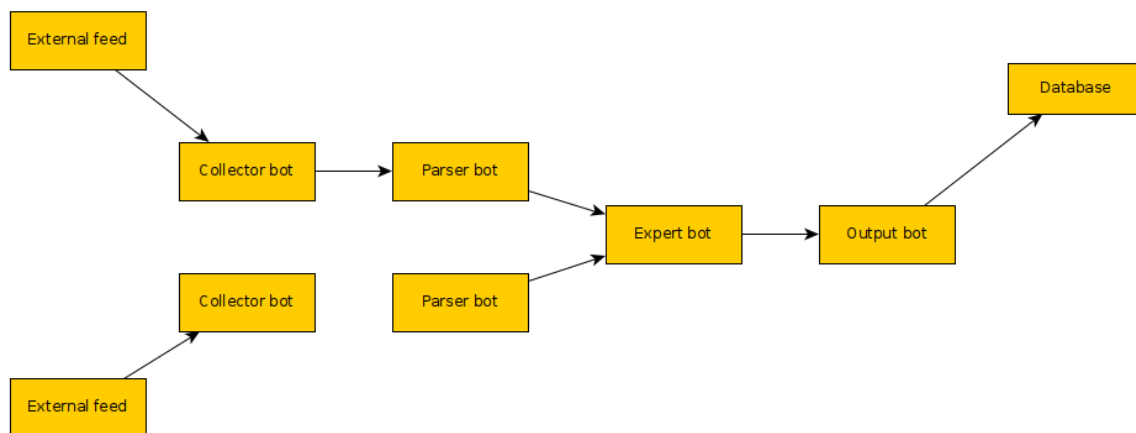


Technically speaking, the bots are mere python scripts running on a single machine and communicating with one another using a Redis³³ broker. This means that you can easily create a new one, if needed.

There are 4 kinds of bots:

- **Collectors:** are used to collect data from a variety of sources - local files, URLs, databases or systems like Shodan³⁴ and MISP
- **Parsers:** are used to gather useful data from raw input like CSV files
- **Experts:** those are the bots used to process and enrich the existing data. They might be used to de-duplicate data or add additional fields, like hostname or geolocation
- **Output:** they are exit nodes that allow us to save the result of the complete pipeline to files, databases or other systems. Usually they accept many known formats and protocols, including popular DB engines, REST API or SMTP.

Figure 3: Example flow of the complete pipeline



5.2 EXERCISE 1 - CREATE A SIMPLE PIPELINE THAT FETCHES DATA FROM A THIRD PARTY AND OUTPUTS IT TO A LOCAL FILE

In this task, we will get familiar with the whole process of how to create bots, make connections between them, and finally run and debug them. We will create a simple pipeline that gathers data from a public third party source (abuse.ch³⁵) and outputs the gathered data to a local file without any additional processing.

In a production environment, we would normally fetch the data from various online sources. However, in this exercise, we want to avoid problems with connections and not up-to-date URLs so we will use a feed hosted locally at path `/opt/enisa/trainings-2019/admin/intelmq/intelmq-clean/shared/ipblocklist.csv`

5.2.1 Enable the installation of IntelMQ

Setup the environment that will run a clean installation of IntelMQ:

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

³³ <https://redis.io/>

³⁴ <https://www.shodan.io/>

³⁵ <https://abuse.ch/>

Now you should be able to access IntelMQ web manager at <http://intelmq.enisa.ex>

If you face server errors (like 503), just wait a few minutes for all the systems to start.

You can now check the status on the “Check” tab. If everything is fine, it should look like this:

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.

5.2.2 Configure the collector

- Choose the *Configuration* tab
- Press the “Add bot” button and place it anywhere on the board.
- From the menu to the left choose *Collector -> File*
- Input the `/opt/shared/ipblocklist.csv` path in *node config* like shown below:

path	<input type="text" value="/opt/shared/"/>	<input type="button" value="⊙"/>
postfix	<input type="text" value="ipblocklist.csv"/>	<input type="button" value="⊙"/>

- Name the feed and data provider (fields “*name*” and “*provider*”) with a custom descriptive name. It will be useful in pipelines with more feeds to easily see the source and type of data in the output.
- Press *OK* to add the bot.

5.2.3 Configure the output

- Create an output node and place it on the board. As the type choose “*File*”
- Configure it to output data to a temporary file at `/opt/shared/out`
- This file will be visible in the 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



Make the connection between the collector and the output:

- Press the “Add queue” button
- Create the connection

Important: remember to always press the **Save configuration** button after making any changes!

5.3 EXERCISE 2 - TEST THE PIPELINE

- Choose the *Management* tab
- Run the pipeline under “*Whole Botnet Status*”
- Check if the output file is being populated

You can see logs of every bot on the *Monitor* tab

All Bots

File-Collector

File-Output

running log

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.



5.4 EXERCISE 3 - ADD PARSER AND EXPERT BOTS

In this exercise we will extract interesting data from a raw feed and sanitise it (remove duplicate entries).

In order to make use of the data that was collected, parser and expert bots must process it. **Parsers** are used to extract specific data from the feed. **Experts** are used to enrich data, e.g. by adding a geolocation tag to IP addresses.

5.4.1 Adding the Parser

Add the *Generic CSV parser*, place it on the board and we will configure it.

As IntelMQ collects the data from different sources in lots of different formats, it must be normalised somehow. For example, the IP address might be described differently depending on the source: "ip", "ip_addr", "ipaddr", "ipaddress", "src_ip" and so on.

In order to provide clearness and uniqueness, a harmonization standard has been created, and all the fields must correspond to it. You can read more about it here:

<https://intelmq.readthedocs.io/en/latest/Data-Harmonization/>

In our case it will be:

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

So configure the "configure" field according to the above:

runtime	runtime	+
column_regex_search	<input type="text" value="{}"/>	⊖
columns	<input style="border: 2px solid red;" type="text" value='["time.source", "destination.ip", "destination.port", "extra.l'/>	⊖
default_url_protocol	<input type="text" value="http://"/>	⊖
delimiter	<input type="text" value=","/>	⊖

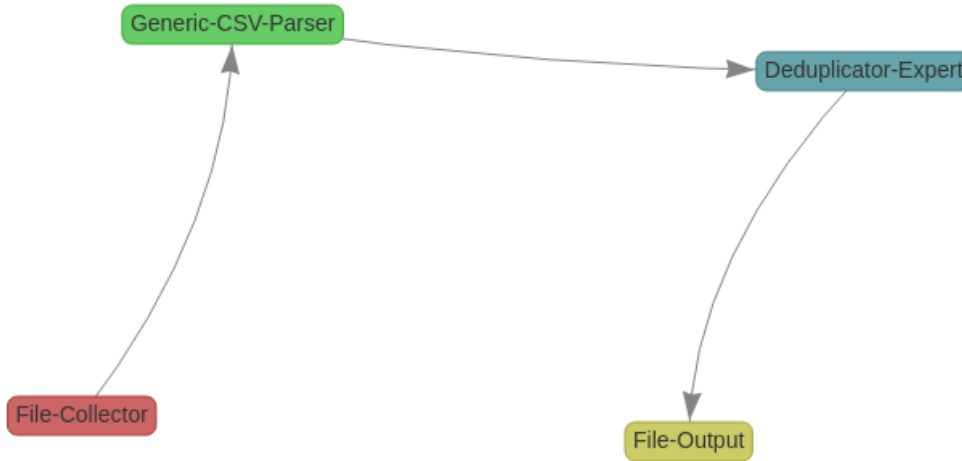
5.4.2 Adding an Expert

Add the de-duplicator expert. The De-duplicator bot takes care not to put the same data to the output twice. You do not have to change anything in the default configuration of it.



5.4.3 Connecting the Bots

Configure the connections like shown below:



5.4.4 Check the Output

Stop the pipeline, clear the output file and rerun the 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": "NjAxOStwOC0wOCxhMDowND01OCw0NS42Ny4yMzEuMTcwLDQ0Ny4yMDE5LTA4LTA4LFRyaWNRQm90DQ0KMjAxNy0wNy0yOCAwMTcyOT0yMyxwNzMuMTJlJE0NS4yMjQsOD0A4MCwsSGVvZG8NCg==",
  "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": "NjAxOStwOC0wOCxhMDowND01OCw0NS42Ny4yMzEuMTcwLDQ0Ny4yMDE5LTA4LTA4LFRyaWNRQm90DQ0KMjAxNy0wNy0yNjAxMDoyMDoyOStwNTcuMTcuMTcyLjIzMCw0NDMsLEhlb2RvDQo=",
  "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": "NjAxOStwOC0wOCxhMDowND01OCw0NS42Ny4yMzEuMTcwLDQ0Ny4yMDE5LTA4LTA4LFRyaWNRQm90DQ0KMjAxNy0wNy0yMyAwNT0zNT0yOStwMTYyOSt0EUNjIuNTQsOD0A4MCwsSGVvZG8NCg==",
  "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"
}
  
```

5.5 EXERCISE 4 - USE MORE COMPLEX COLLECTOR AND OUTPUT BOTS

Now we are ready to create our final pipeline.

We will collect the data from our locally hosted honeypot, process it using the pipeline we created and finally save the results to Elasticsearch. The Analysts can browse the results conveniently and they can be visualised using Kibana. One of the main advantages of IntelMQ is aggregating data from multiple feeds and saving them in Elasticsearch under one index.

5.5.1 SNARE/TANNER honeypot



As the input, we will use the honeypot consisting of two parts working together - **TANNER** and **SNARE**.

SNARE is a honeypot endpoint, you can use it to clone any website and present it to potential attackers. However, the full analytic logic is placed in TANNER. It contains many configurable modules that allow emulating typical web vulnerabilities (XSS, SQL Injection etc.).

In a model like this we can have multiple SNARE endpoints (e.g. for different websites) with common logic implemented in a central TANNER instance.

In our VM, the honeypot is already configured and running at <http://honeypot.enisa.ex>. There is a script sending malicious requests every few seconds at `/opt/enisa/trainings-2019/admin/intelmq/scripts/send.py`

You can run it now:

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

By default, honeypot logs are saved in `/opt/enisa/trainings-2019/admin/intelmq/intelmq-clean/shared/snare.log`

You can look at this file now to see how it is being populated.

5.5.2 Adding a custom bot

There is no default parser bot in IntelMQ that understands SNARE's log format. Luckily, it is very easy to create and add a custom one. In our instance a custom bot is already added, you can read its source code at:

```
/opt/enisa/trainings-2019/admin/intelmq/bots/parsers/snare/parser.py
```

You can read more about creating custom bots here:

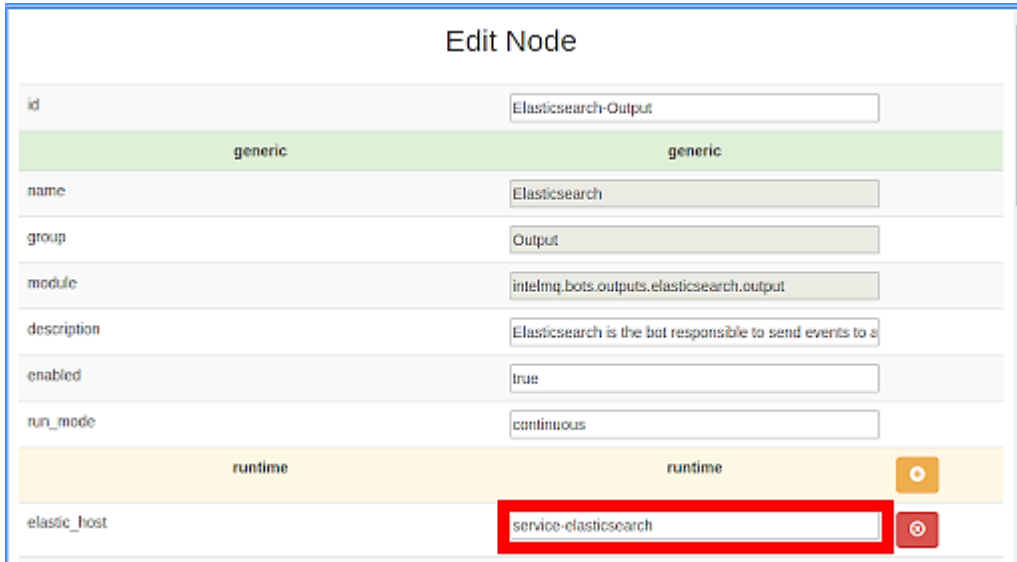
<https://intelmq.readthedocs.io/en/latest/Developers-Guide/#bot-developer-guide>

Now we are ready to create the complete pipeline.

- As the input file put the path `/opt/shared/snare.log` (remember to name both feed and provider correctly!).
- As the Parser-bot use `SNARE` - our custom created one.
- Add the de-duplicator, just like in previous the task.
- As the output, we will use Elasticsearch. Choose the Elasticsearch Output-bot and configure it as shown below:



The `elastic_host` should be “`service-elasticsearch`”.



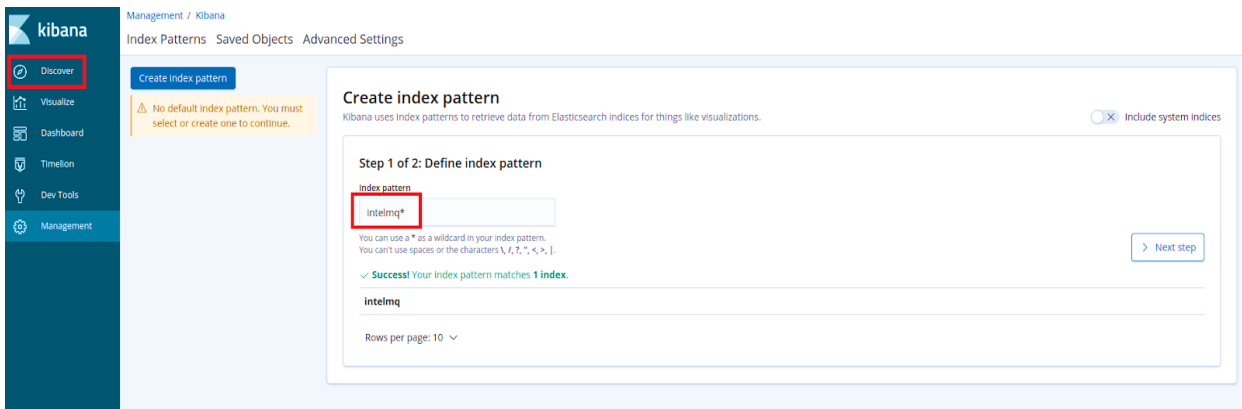
generic	
id	Elasticsearch-Output
name	Elasticsearch
group	Output
module	intelmq.bots.outputs.elasticsearch.output
description	Elasticsearch is the bot responsible to send events to e
enabled	true
run_mode	continuous

runtime	
elastic_host	service-elasticsearch

Save the configuration and run the pipeline.

If everything worked fine you should be able to see the results at kibana.enisa.ex (if you see 503 errors shortly after starting the exercise just wait a few minutes for the environment to fully set up).

In Kibana click the “*Discover*” tab and create an index pattern named “*intelmq*”:



Management / Kibana
Index Patterns Saved Objects Advanced Settings

Discover

Create index pattern

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
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

> Next step



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

Time	_source
November 19th 2019, 18:12:45.000	<pre>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: eyJzb3VyY2UudXJsIjogImh0dHA6Ly9ob25leXBvdC8lCAidGltZS5zb3VyY2UiO1A1MjAxOS0wOS0yNVQxMzoyNz00CswMDowMCIsICJzb3VyY2UuaXAiOiAiMzcuMjAxLjIwNi4yMzIiLCAiZkh0cmEucEuc6FyYWIzIjoge319 feed.url: file://localhost/opt/shared/snare_log.json feed.accuracy: 100 _id: IY0mhG4BhV5jksuX-Z60 _type: events _index: intelmq _score: -</pre>
November 19th 2019, 18:12:45.000	<pre>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: eyJzb3VyY2UudXJsIjogImh0dHA6Ly9ob25leXBvdC9pbmRleCIsICJ0aW11LnNvdXJjZSI6ICImDE5LTASLTI1VDEzOjMwOjM2KzAwOjAwIiwgInNvdXJjZS5pcCI6ICIxMDcuMjAxLjIwNi4yMzIiLCAiZkh0cmEucEuc6FyYWIzIjoge319 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: I40mhG4BhV5jksuX-dEG _type: events _index: intelmq _score: -</pre>
November 19th 2019, 18:12:45.000	<pre>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: eyJzb3VyY2UudXJsIjogImh0dHA6Ly9ob25leXBvdC9pbmRleCIsICJ0aW11LnNvdXJjZSI6ICImDE5LTASLTI1VDEzOjMwOjM5KzAwOjAwIiwgInNvdXJjZS5pcCI6ICIxODUuNTUuMjYyLjE5NCIsICJleHRyYS5wYXJhbXMiOiB7ImxvZ2luIjogIiBcIiBvc1AxPTEtLSIsICJzdWJtaXQ1OiAiIFN1Ym1pdCIsICJwYXNzd29yZCI6ICImTIzMTIzIn19 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: Jf0mhG4BhV5jksuX-dEx _type: events _index: intelmq _score: -</pre>

You can see the complete pipeline ready in the *intelmq-populated* environment:

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



6. THEHIVE ADMIN

6.1 INTRODUCTION:

Parameter	Description	Duration
Main Objective	This exercise introduces TheHive – a platform supporting incident handling. Trainees are going to configure a TheHive instance and setup integration with other tools, including Cortex, Elasticsearch and MISP.	-
Targeted Audience	The exercise is dedicated to (new) CSIRT staff involved in incident handling and tools administration.	
Total Duration	1,5 hours	90 minutes
Time Schedule	Introduction to the exercise	15 minutes
	Task 1: Setup TheHive & Cortex accounts	10 minutes
	Task 2: Configure Cortex analysers	10 minutes
	Task 3: Configure the Hive-Cortex integration	10 minutes
	Task 4: Configure the Hive-MISP integration	15 minutes
	Task 5: Creating custom Cortex analyser	15 minutes
	Task 6: Report templates, Case templates, Dashboards	15 minutes

In this part of the exercise, you will be introduced to TheHive³⁶ – a platform for incident handling dedicated for Security Operational Centres. TheHive provides an efficient platform for multiple users to investigate cases in parallel. The software has built-in tools for data enrichment and automatically correlates tags and observables. You will learn about the components like Cortex and analysers. We will also synchronize TheHive with MISP³⁷.

TheHive uses Elasticsearch as its database. In the training environment, the Elasticsearch instance used by TheHive is storing its files on another Kubernetes³⁸ container. Such a setup allows restarting TheHive container without losing data (that normally happens to all changes that were made inside the container).

Cortex³⁹ is the environment for small worker applications called **analysers**. These applications can be invoked in a number of ways – from TheHive, from the Cortex web interface (using the Cortex REST API) or using the Cortex4py library. Many analysers come shipped with Cortex, but it is very easy to create new ones using any programming language.

6.2 TASKS:

To start the learning environment, execute following commands once you boot the virtual machine (VM user: enisa, password: enisa):

³⁶ <https://thehive-project.org>

³⁷ <https://www.misp-project.org>

³⁸ <https://kubernetes.io>

³⁹ <https://github.com/TheHive-Project/CortexDocs>



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

Followed by:

- `./start_excercise.sh` (pass: enisa)

Now wait for your environment to come up and get ready, as shown in the following screenshot:

```
==> v1/Pod(related)
NAME                               AGE
hive-elastic-84c7d478f4-jl28x     14s
ideal-echidna-misp2-5d7fc88b9c-xvgbs 14s
thehive-5cb485c7bb-9f952          14s
thehive-cortex-58777c676-t82h6     14s

==> v1/Service
NAME           AGE
cortex-service 14s
elastic-service 14s
misp2-service  14s
thehive-service 14s

==> v1beta1/Deployment
NAME           AGE
ideal-echidna-misp2 14s

==> v1beta1/Ingress
NAME           AGE
ideal-echidna-misp2 14s
ingress-cortex  14s
ingress-thehive  14s

Your environment is up and ready!
```

If you want to shutdown the exercise environment, execute

- `./stop_excercise.sh`

If you want to start all over again you can execute

- `./stop_excercise.sh` and then `./start_excercise.sh`.

This will erase all progress that you have made and set everything to the initial state.

6.2.1 Setup accounts

We now need to setup the admin accounts for both instances.

- Open TheHive instance web UI at thehive.enisa.ex.
- Click “Update database” and then set up the admin account (Note: if you encounter an SSL warning, you can ignore it as it's a training environment).



- On the first login, TheHive needs to build databases and create initial admin credentials. It is important to note down the new password or pick something easy to remember. For simplicity you can go with `admin : admin`.
- Now you are ready to login to TheHive instance. Feel free to get familiar with the graphical user interface.
- Open the Cortex instance web UI at [cortex.enisa.eu](https://cortex.enisa.europa.eu)
- Click “Update database” and set up the admin account. For simplicity you can go with `admin : admin`.
(Note: if you encounter an SSL warning, you can ignore it as it is a training environment).

Then login to the graphical user interface and create a new organization:

- Click the “+ Add organisation” button, and name it “enisa.ex”).
Note: this step is necessary because the default “cortex” organization can contain only administrative accounts.

Then create a new user in the new organisation *enisa.ex*:

- Navigate to Users -> + Add user and give this new user *read*, *write*, and *orgadmin* access for the use of TheHive. You must set a password for it, use the following credentials: login: `admin.enisa.ex` and password: `admin`.

6.2.2 Configure Cortex analysers

The next step is to log out and then again log in to a freshly created account (suggested credentials were `admin.enisa.ex : admin`).

Once logged in, select and enable some analysers e.g. Maxmind GeoIP⁴⁰:

- Organization -> Analyzers ->MaxMind_GeoIP_3_0 -> click Enable -> Save

This is also the right place to configure analysers (eg. placing login:password, api keys...).A good example of a useful Cortex analyser is *MISP_2_0*.

It allows searching for observables/attributes in a MISP instance for which you provide the URL and the API-key.

You can check if the analyser works correctly by clicking on:

- “New analysis” -> Data type : ip -> Data: 195.187.6.2
- Tick “MaxMind_GeoIP_3_0” -> Start.

You can check the results by clicking on “View” on a suitable row.

⁴⁰ <https://www.maxmind.com>



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,
```

6.2.3 Configure TheHive - Cortex integration.

Cortex - TheHive integration is one of the key elements of this part of the training. It allows Security Analysts to easily enrich information gathered in the course of investigations in order to better understand what happened.

To do that, you need to obtain the API- key of the newly created user:


- Organization -> Users -> Create api key -> Reveal

Next, replace it in the *application.conf* file in the *thehive-config* directory (section cortex, field key). Usually, a config file is in the *“etc/thehive/application.conf”* path.

You also have to uncomment line `play.modules.enabled += connectors.cortex.CortexConnector`

Then restart TheHive container by executing the *./restart_thehive.sh* script, then wait a few seconds.

Now go back to TheHive instance, and check if the integration works by going to <name of the user> button -> about, and checking Cortex status. Integration is also indicated by a Cortex icon in a green circle at the bottom-right corner of the TheHive GUI.

Version: 3.3.0-1 



To check if the Cortex analyser works, follow these steps:

- Create a New Case and add an Observable
- E.g. IP: 195.187.6.2
- Run MaxMind_GeoIP analyser by clicking on the observable, followed by clicking on the red icon in the Analysis area.
- Next, check the result by clicking on a date of analysis -> Show raw report.

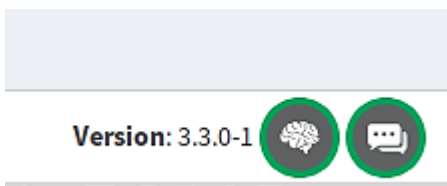
6.2.4 Configure the Hive-MISP integration and check if alerts are fetched

Similarly to Hive-Cortex integration, the process of enabling integration comes down to getting the API- key of a MISP user (in this case it is `gxPEOFh04jGZriMUhBI3U9lyOp7lrxKYiflDMMB3`) and putting it in the `application.conf` file in the `thehive-config` directory (section MISP, field "key") as well as uncommenting `play.modules.enabled += connectors.misp.MispConnector`.

Then restart the TheHive pod, just like in the previous task.

When restarted, the integration should work and this will be indicated with a green circle around the MISP icon in the bottom-right corner of the TheHive web-UI. Moreover, the web-UI will show up the number of events fetched from MISP in the Alerts tab.

Now go to the Alert tab and check the example alert created from a MISP event.



Note: in this exercise we are using `play.ws.ssl.loose.acceptAnyCertificate = true` in order to make integration possible even if the MISP instance has a self-signed certificate. In a production environment, that setting is not recommended of course!

6.2.5 Creating a custom Cortex analyser

The goal of this task is to create a custom analyser for data in Elasticsearch and by doing so, integrating those two platforms. We are going to query for data pushed to Elasticsearch by an IntelMQ pipeline.

To achieve that, we need to create at least two files:

- One `.json` file with the specifications of the analyser;
- A second one with the python code of the analyser itself.
- There is an optional third file named `requirements.txt` with a list of required packages/libraries to run the analysers.



Now go to `/opt/enisa/trainings-2019/admin/thehive/cortex-analyzers` and see the list of installed analysers.

Then navigate to the ESlookup directory and look at the file `eslookup.json`. In there you can find the definition of the analyser e.g. name, version, author, accepted input data types and location of python script to execute.

A definition file can also contain information about the maximum TLP level of the observable that can be passed to the analyser. This can help prevent accidental leaks of sensitive data by unsuspecting users.

Finally, look at the analyser's code file `eslookup.py`.

Try to figure out what it is doing. When you're done, go to the Cortex interface and enable it (just like you did before with MaxMind (6.2.2), you will find it under the name "ES data lookup_1_0". This is also the right moment to enable the IP_ASN analyser.

Try the new custom analyser against two IP addresses: 108.185.19.99 and 122.15.121.100, note the difference in output.

Job report

Report

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

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": []
}
```



6.2.6 Responders

The basic idea behind responders is the same as for analysers - they get data and provide some useful actions. The main difference is that analysers are run against particular observables; responders are run against cases or alerts. In addition, analysers provide you with some additional data; while responders can trigger some actions like creating a ticket, sending an email etc.

Example responders are available here:

<https://github.com/TheHive-Project/Cortex-Analyzers/tree/master/responders>

6.2.7 Report templates

Output from analysers may be customized using report templates. They allow showing results using html/bootstrap instead of plain json.

To add a report template designed to our custom analyser, go to:

- TheHive -> Admin -> Report templates
- In ES_data_lookup_1_0 and column “Long template”, click on “Default template” to modify it.

Then paste the following:

- `Found in log db!`
- `Not found in log db!`

The above code checks the Boolean value of the “Summary” field in the json returned by our custom analyser.

- Click on “Save template”.

Now go to the test case and see if the report changed for the two IP addresses (108.185.19.99 and 122.15.121.100). Remember, you have to add them as observables first.

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

Found in log db!

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

Not found in log db!

6.2.8 Case templates

Let us assume that for the needs of particular organisation we need to know the number of people affected by a particular incident. To represent that information, we will create a Case custom field (Admin -> Case custom fields) and fill the form, e.g.

- Name: number of people affected
- Description: number of people affected by that incident
- Type: number
- Click on “Save field”

Then go to “Case template management” (Admin -> Case templates) and fill in the form:

- Template name: company X
- Title prefix: X-
- Description: this template concern cases related to company X



- Custom fields -> Add a custom field -> Select “number of people affected” from the dropdown menu
- Click on Save case sample.

Now when you create a new case, you will be able to pick the newly created template and use additional data fields.

6.2.9 Dashboards

Dashboards are a very handy way of visualising data in TheHive. You can go to the Dashboards tab and see default ones.

We will create a new custom dashboard based on the “number of people affected” values from our cases.

To do that, go to the Dashboards tab and next:

- Click on “Create new Dashboard”, give it a name and description
- Select Visibility to “Shared” and click “Create”.
- In the new window on the right side, select “Donut” as a type of graph,
- Drag&Drop it to the empty place in the middle.
- In the pop-up window in the entity dropdown select “Case” and in “Aggregation Field” select “customFields.numberPeopleAffected.number”.
- Next, click “Apply” and “Save”.

Your custom dashboard is now ready. Now you can add some test cases to see how it looks.





ABOUT ENISA

The mission of the European Union Agency for Cybersecurity (ENISA) is to achieve a high common level of cybersecurity across the Union, by actively supporting Member States, Union institutions, bodies, offices and agencies in improving cybersecurity. We contribute to policy development and implementation, support capacity building and preparedness, facilitate operational cooperation at Union level, enhance the trustworthiness of ICT products, services and processes by rolling out cybersecurity certification schemes, enable knowledge sharing, research, innovation and awareness building, whilst developing cross-border communities. Our goal is to strengthen trust in the connected economy, boost resilience of the Union's infrastructure and services and keep our society cyber secure. More information about ENISA and its work can be found at www.enisa.europa.eu.

ENISA

European Union Agency for Cybersecurity

Athens Office

1 Vasilissis Sofias Str
151 24 Marousi, Attiki, Greece

Heraklion office

95 Nikolaou Plastira
700 13 Vassilika Vouton, Heraklion, Greece

enisa.europa.eu



ISBN 000-00-0000-000-0
doi: 0000.0000/000000