



EUROPEAN UNION AGENCY
FOR CYBERSECURITY



ORCHESTRATION OF CSIRT TOOLS

TRAINERS HANDBOOK

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.	EXECUTIVE SUMMARY	3
1.	INTRODUCTION	8
1.1	AIM OF THIS TRAINING	8
1.2	STRUCTURE OF THE TRAINING	8
1.3	TOOLS USED IN THIS TRAINING	9
1.3.1	MISP	9
1.3.2	TheHive	9
1.3.3	Cortex	9
1.3.4	IntelMQ	9
1.3.5	Elasticsearch	9
1.3.6	Kibana	9
1.3.7	SNARE and TANNER	9
1.4	ARCHITECTURE	9
2.	ADMIN PART - GENERAL INFORMATION	10
2.1	INTRODUCTION	10
2.2	CREDENTIALS	10
3.	MISP ADMIN	12
3.1	INTRODUCTION	12
3.2	PRECONFIGURED STATES	13
3.2.1	misp-bare	13
3.2.2	misp-configured	13
3.3	EXERCISE	13
3.3.1	Preparation	13
3.3.2	Events	15
3.3.3	Galaxies	16
3.3.4	Taxonomies	16
3.3.5	Adding users	17
3.3.6	Organisation	18
3.3.7	Role permissions	19
3.3.8	Dashboard and Statistics	19
3.3.9	Automation API	20
3.3.10	Additional configuration	20
3.3.11	Synchronisation	20

4.	ELASTICSEARCH ADMIN	22
4.1	INTRODUCTION	22
4.2	PRECONFIGURED STATES	22
4.2.1	elasticsearch-bare	22
4.3	EXERCISE: ELASTICSEARCH BASIC ADMINISTRATION	23
4.3.1	Overview of Elasticsearch	23
4.3.2	Overview of Kibana	25
4.3.3	Configure the exercise	26
4.4	GET FAMILIAR WITH ELASTICSEARCH	28
4.4.1	Create an index	28
4.4.2	Adding data to the cluster	31
4.4.3	Health monitoring	34
4.4.4	Bulk insert more test data	35
4.4.5	Exercise: find interesting data in the cluster.	35
4.5	GET FAMILIAR WITH KIBANA	35
4.5.1	Configure index for dashboards	35
4.5.2	Use Kibana to discover your data.	37
4.5.3	Exercise: find interesting data in the cluster.	38
4.5.4	Create a visualisation	38
4.5.5	Exercise: create your own visualisation	41
4.5.6	Real time visualisations	41
5.	INTELMQ ADMIN	41
5.1	INTRODUCTION	41
5.1.1	Pipeline	41
5.1.2	Bots42	
5.2	EXERCISE 1 - CREATE A SIMPLE PIPELINE THAT FETCHES DATA FROM A THIRD PARTY AND OUTPUTS IT TO A LOCAL FILE	42
5.2.1	Enable the installation of IntelMQ	43
5.2.2	Configure the collector	43
5.2.3	Configure the output	43
5.3	EXERCISE 2 - TEST THE PIPELINE	44
5.4	EXERCISE 3 - ADD PARSER AND EXPERT BOTS	45
5.4.1	Adding the Parser	45
5.4.2	Adding an Expert	45
5.4.3	Connecting the Bots	46
5.4.4	Check the Output	46
5.5	EXERCISE 4 - USE MORE COMPLEX COLLECTOR AND OUTPUT BOTS.	46
5.5.1	SNARE/TANNER honeypot	47
5.5.2	Adding a custom bot	47

6.	THEHIVE ADMIN	49
6.1	INTRODUCTION:	49
6.2	TASKS:	50
6.2.1	Setup accounts	51
6.2.2	Configure Cortex analysers	52
6.2.3	Configure TheHive - Cortex integration.	53
6.2.4	Configure the Hive-MISP integration and check if alerts are fetched	54
6.2.5	Creating a custom Cortex analyser	54
6.2.6	Responders	56
6.2.7	Report templates	56
6.2.8	Case templates	56
6.2.9	Dashboards	57
7.	ANALYSTS PART - GENERAL INFORMATION	58
7.1	INTRODUCTION	58
7.2	CREDENTIALS	58
8.	MISP ANALYST	59
8.1	INTRODUCTION	59
8.2	PRECONFIGURED STATES	59
8.2.1	misp-bare	59
8.2.2	misp-configured	59
8.3	PREPARATION	60
8.3.1	Environment setup	60
8.3.2	Resetting your progress	60
8.3.3	Login in MISP1	60
8.3.4	Events	60
8.3.5	Adding events	61
8.4	EXERCISES	62
8.4.1	Exercise 1 - Adding an Event	62
8.4.2	Exercise 2 - Search and Correlation	65
8.4.3	Galaxies	67
8.4.4	Taxonomies	67
9.	LOGS ANALYSIS ANALYST	68
9.1	INTRODUCTION	68
9.2	OVERVIEW OF INTELmq	68
9.3	CONFIGURE THE EXERCISE	70
9.3.1	Ensure that DNS is configured properly	70
9.3.2	Apply the helm configuration file	70



9.3.3	Completion of the installation	70
9.3.4	Ensure that Elasticsearch works correctly	70
9.3.5	Ensure that Kibana works correctly.	71
9.4	GET FAMILIAR WITH INTEL MQ	72
9.4.1	Get familiar with the pipeline	72
9.4.2	Start the botnet	73
9.4.3	Familiarise yourself with the honeypot	74
9.4.4	Take a look at the data	75
9.4.5	Exploit hunt	80
9.4.6	Exercise 1	81
9.4.7	Exercise 2	81
9.4.8	Exercise 3	85
10.	THEHIVE ANALYST	86
10.1	INTRODUCTION:	86
10.2	TASKS:	86
10.2.1	General workflow	87
10.2.2	Let's get into UI!	88
10.2.3	Hands-on	88
11.	ARCHITECTURE AND TECHNICAL BACKGROUND	90
11.1	INTRODUCTION	90
11.2	ARCHITECTURE	91
11.3	ADDING A NEW SYSTEM	91
11.4	ADDING A NEW SCENARIO	92
11.5	DEBUGGING ESSENTIALS	93
11.6	POSSIBLE ERROR MESSAGES:	95
11.6.1	Error type 1	95
11.6.2	Error type 2	95
11.6.3	Error type 3	95
11.6.4	Error type 4	95
12.	BIBLIOGRAPHY/REFERENCES	96
A	ANNEX: EXAMPLE OF ANNEX	96
A.1	ANNEX SUBSECTION	96
A.2	ANNEX SUBSECTION	96
A.2.1	Annex Second Subsection	96
A.2.2	Annex Second Subsection	96



EXECUTIVE SUMMARY

This material contains an update to the existing ENISA Collection of CSIRT trainings, specifically focusing on the trainings labelled under “Technical” and “Operational” on the ENISA CSIRT training webpages². The revised and renewed training materials are based on good practices and include methodologies, tools and procedures to be compliant with the “Train the Trainer” approach.

The updated scenarios also include content that is in line with the current technologies and methodologies in the EU-wide domain of automation and orchestration in Incident Response. The training includes performance indicators and means, supporting those who use it to increase their operational competence.

The new training material presented in this document consist of multiple independent modules, each covering a particular combination of open-source tools that are widely accepted and recognised within the CSIRT and cybersecurity communities. They are designed to help CSIRTs cope with today’s large amount of valuable information and data sources and to facilitate sharing valuable information to other teams and communities. This allows the CSIRT to enrich data with their own intelligence and then share it back to the wider CSIRT community, a powerful weapon in today’s battle against large scale and sophisticated threats.

The underlying technical framework developed for the training allows modifying and extending the materials to adapt to the fast evolving landscape of CSIRT tooling, making the training reusable and future-proof. This concept aims to give both new and experienced teams the opportunity to “test-drive” new tools and explore how they can be integrated in existing setups in a more efficient way. The technical platform is provided as a Virtual Machine and it was conceived to allow a smooth transition to cloud based hosting.

For each interesting combination of the included tools, one part of the training is intended for staff that will setup, configure and maintain the tools. The second part is more aimed towards the security analysts that will be using these combined tools during their daily activities. The two parts are not exclusive so staff that is involved in both types of duties can participate in both parts.

It is worth noting that the technical framework allows all the different training modules to be setup independently from each other. The second (analyst) part can be deployed without first having to go through the administration part.

The duration of the whole training is estimated for two days (approximately 16 hours), including breaks, but it can also be used only with a specific subset of the training modules, and allow shortening the required time needed to go through the training and to adapt the training to their specific needs.

The updated material consists of a Virtual Machine that allows deployment of all the required training scenarios, a trainers Handbook, students’ Toolsets and slides to accompany the trainer and the students while executing exercises. The practical approach applied in this material leads the trainees through scenarios based on simulated situations and mirroring typical CSIRT processes.

The training scenarios are targeting mainly technical CSIRT staff (both administrators and security analysts) who would like to improve their skills, effectiveness and cooperation with other teams and stakeholders.

² <https://www.enisa.europa.eu/topics/trainings-for-cybersecurity-specialists/online-training-material>

INTRODUCTION

1.1 AIM OF THIS TRAINING

The training aims in supporting new as well as more experienced CSIRTs teams in orchestrating multiple open-source tools for information collection, processing and exchange to allow automation of common data handling and analysis tasks.

The underlying technical framework developed for the training allows modifying and extending the materials to adapt to the fast evolving landscape of CSIRT tooling, making the training reusable and future-proof. This concept aims to give both new and experienced teams the opportunity to “test-drive” new tools and explore how they can be integrated in existing setups in a more efficient way. The technical platform is provided as a Virtual Machine (VM) and it was conceived to allow a smooth transition to cloud based hosting.

It is worth noting that the technical framework allows all the different training modules to be setup independently of one another³.

There is also a Chapter (11) devoted to the architecture and the technical background of the training. It includes debugging tips and sections on how to add new tools and/or scenarios to the platform.

1.2 STRUCTURE OF THE TRAINING

The training is divided into two parts, each with a different target group in mind. Needless to say of course that this division is not exclusive. In many cases, CSIRT staff take up technical system administrator duties while they also work as security analysts in their team.

The first part is dedicated to the technical aspects of the orchestration, allowing to practice with a selection of open-source tools that are presented below. This part is more aimed towards system administrators or CSIRT staff that will be responsible for the setup and maintenance of the tools.

The second part – mainly intended for security analysts - deals with analytical workflows, focusing on leading simple analysis designed as training scenarios. Each of the scenarios demonstrates how different tools can facilitate a typical CSIRT workflow. The emphasis is placed on the benefits of having multiple CSIRT tools interconnected (orchestrated) and supporting analysts' work.

Each part consists of separate blocks that can be run independently of each other in any order of the choice. This allows tailoring the training exactly towards the needs of individuals or teams. How this can be achieved, is described further in this Handbook.

Training materials for both parts consist of a Trainers Handbook, students' Toolsets and slides that take you through the training systematically. A Virtual Machine allows deploying all the available scenarios.

They can be downloaded from the ENISA website here:

<https://www.enisa.europa.eu/topics/trainings-for-cybersecurity-specialists/online-training-material/technical-operational/#Orchestration>

The training infrastructure is based on state of the art open-source containerisation and orchestration technologies

³ E.g., the analyst part can be deployed without first having to go through the administration part.



1.3 TOOLS USED IN THIS TRAINING

The following open-source tools have been chosen for this training. They are commonly used in CSIRTs and other security teams in general for monitoring, information collection, data & incident analysis and information exchange.

1.3.1 MISP

MISP⁴ (Malware Information Sharing Platform) is the leading open-source threat intelligence platform, with a very high adoption rate in CSIRTs across the EU and worldwide. MISP has a built-in distributed sharing mechanism that is used to exchange Indicators of Compromise (IoCs) between multiple teams.

1.3.2 TheHive

TheHive⁵ is an incident-handling platform, supporting common analysts' workflows. It supports the management of structured and unstructured information, automation and collaboration between team members. TheHive integrates well with other systems commonly used by CSIRTs and other IR teams.

1.3.3 Cortex

Cortex⁶ is a companion tool for TheHive that is used for correlation, enrichment of observables and automation of response actions.

1.3.4 IntelMQ

IntelMQ⁷ is a modular system for fully automated collection, normalization, enrichment and distribution of data feeds.

1.3.5 Elasticsearch

Elasticsearch⁸ is a versatile and general-purpose solution for data storage, popular for storing logs and other information relevant for incident detection and analysis. It is highly scalable and has rich query capabilities.

1.3.6 Kibana

Kibana⁹ is a web frontend for Elasticsearch, allowing analysts to query and visualize data, including creating helpful real-time dashboards based on gathered data.

1.3.7 SNARE and TANNER

SNARE¹⁰ is a web honeypot for monitoring of incoming attacks. TANNER¹¹ is used to control the behaviour of multiple SNARE instances and aggregate the information that is collected.

1.4 ARCHITECTURE

The training infrastructure is based on state of the art open-source containerisation and orchestration technologies such as Kubernetes¹² and Helm¹³. This approach allows simplifying further development of the materials in the future by adding new tools and rearrange existing ones.

Moreover, the solution can be adapted to work natively in a hosted cloud infrastructure, removing the need for local setup of the environment and thus completely streamlining the training process.

⁴ <https://www.misp-project.org/>

⁵ <https://thehive-project.org/>

⁶ <https://github.com/TheHive-Project/CortexDocs>

⁷ <https://github.com/certtools/intelmq>

⁸ <https://www.elastic.co/products/elasticsearch>

⁹ <https://www.elastic.co/products/kibana>

¹⁰ <http://mushmush.org/>

¹¹ <http://mushmush.org/>

¹² <https://kubernetes.io/>

¹³ <https://helm.sh/>

More information on the architecture and technical setup of the training can be found in Chapter 11 of this handbook.

We encourage trainers to have a look at the section with Debugging Essentials (11.5) at the end of this Handbook since the information in that section might come in handy when facing technical difficulties while deploying the training scenarios.

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 VirtualBox¹⁴ 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	-	-

¹⁴ 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 MISP¹⁵ – 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 website¹⁶.

¹⁵ <https://www.misp-project.org/>

¹⁶ <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 VirtualBox¹⁷ 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
```

¹⁷ <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 (8) 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 levels¹⁸:
 - **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

¹⁸ <https://www.first.org/ttp/>

3.3.6 Organisation

Each user 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:

- 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

A brief description of the organisation

A description of the organisation that is purely informational.

The following fields are all optional.

Nationality

Not specified

Sector

For example "financial".

Type of organisation

Freetext description of the org.

Contacts

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

- **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

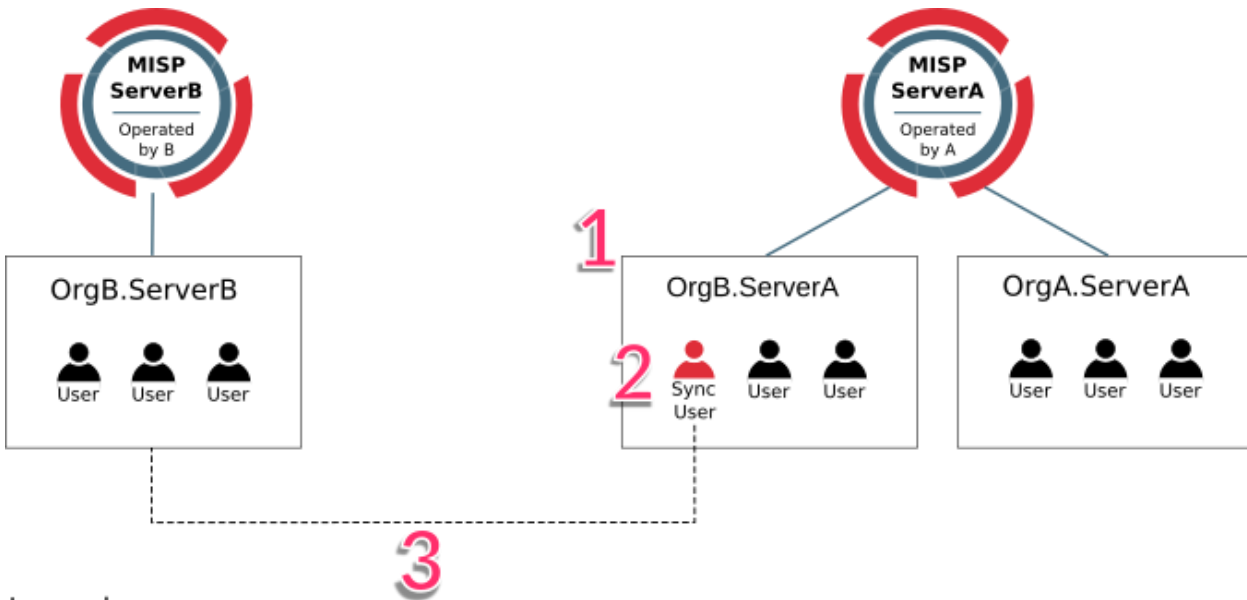
Other MISP settings can be changed later (e.g. plugin options, Redis server configuration for caching, etc...). More information on these settings and configurations can be found in the official MISP documentation¹⁹.

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)

¹⁹ <https://www.misp-project.org/documentation/>

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.

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.

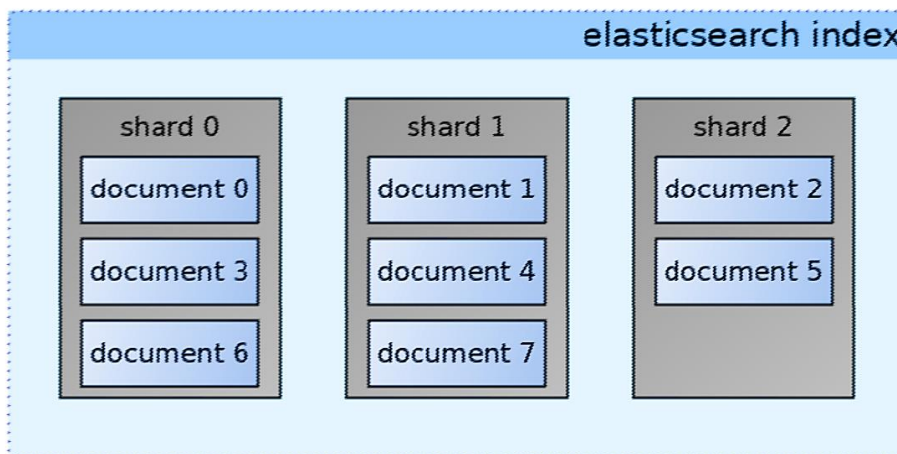
4.3 EXERCISE: ELASTICSEARCH BASIC ADMINISTRATION

4.3.1 Overview of Elasticsearch

Elasticsearch²⁰ is a distributed search and analytics engine designed for fast full text and structured search. Elasticsearch exposes a REST²¹ API²² that can be used directly by CURL²³, 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 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

²⁰ <https://www.elastic.co/>

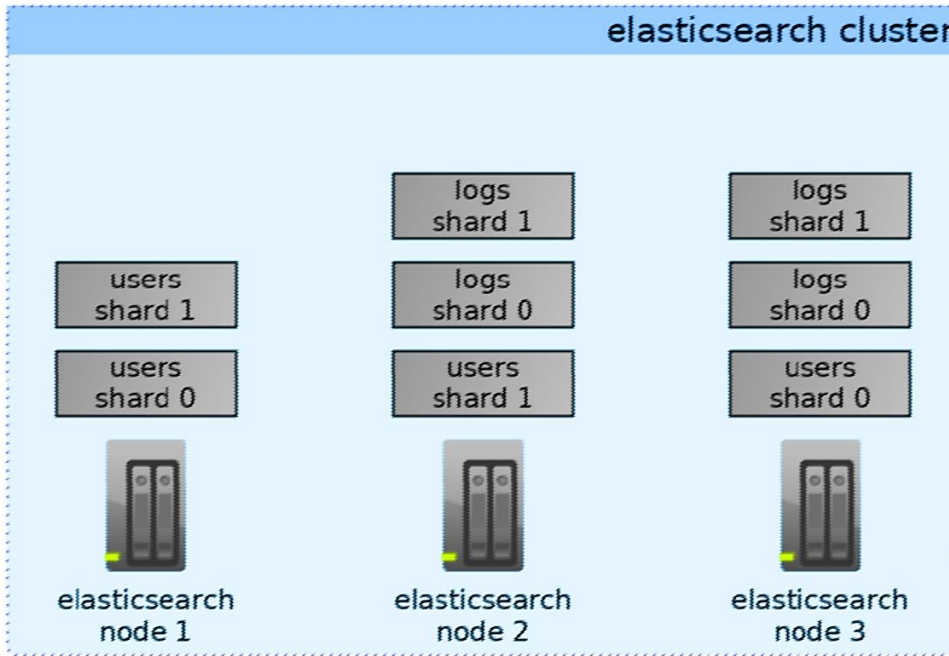
²¹ REST: Representational state transfer. It's a style of creating HTTP services, especially popular for API-heavy application

²² API: Application Programming Interface - interface exposed by the application for the programmers

²³ curl: <https://curl.haxx.se/>. Command line tool for sending requests using multiple supported protocols. Most commonly used with HTTP

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 Kibana²⁴. 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.

²⁴ <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",
```



```

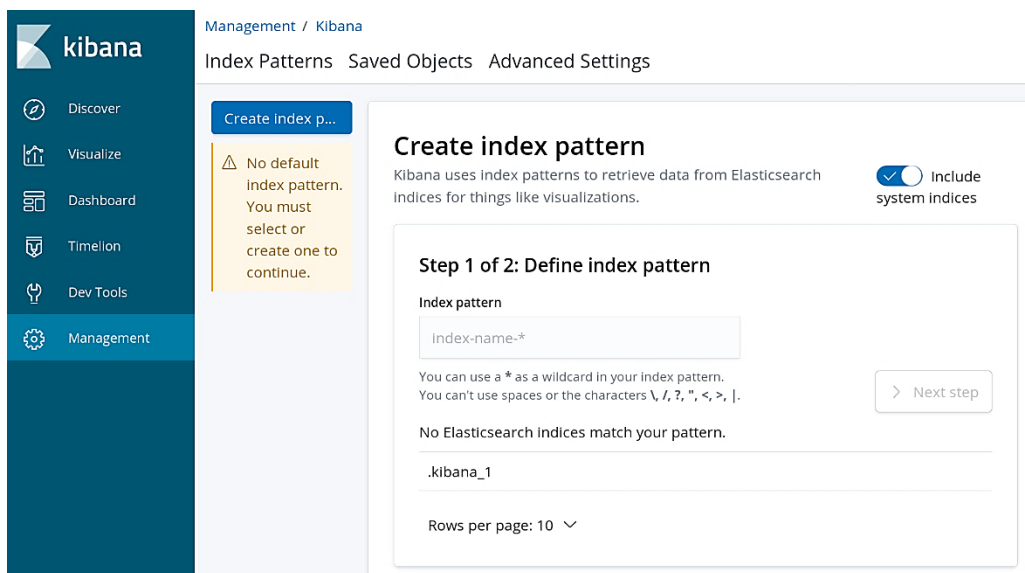
"build_type" : "tar",
"build_hash" : "1fd8f69",
"build_date" : "2019-02-13T17:10:04.160291Z",
"build_snapshot" : false,
"lucene_version" : "7.6.0",
"minimum_wire_compatibility_version" : "5.6.0",
"minimum_index_compatibility_version" : "5.0.0"
},
>tagline" : "You Know, for Search"
}

```

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 UNIX²⁵ 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"
        }
      }
    }
  }
}
```

²⁵ <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 depreciation 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 Nagios²⁶ or Icinga²⁷. 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).

²⁶ <https://www.nagios.org/>

²⁷ <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 indices for things like visualizations. 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 indices for things like visualizations. 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	Searchable	Aggregatable	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.

kibana
15,126 hits
New Save Open Share Inspect
🔄 Auto-refresh
🕒 Last 5 years

Options Refresh

🔍 Discover
📊 Visualize
📄 Dashboard
🕒 Timelion
🔧 Dev Tools
⚙️ Management

logs*

Selected fields

? **_source**

This field is present in your Elasticsearch mapping but not in the 500 documents shown in the doc table. You may still be able to visualize or search on it.

Available fields >

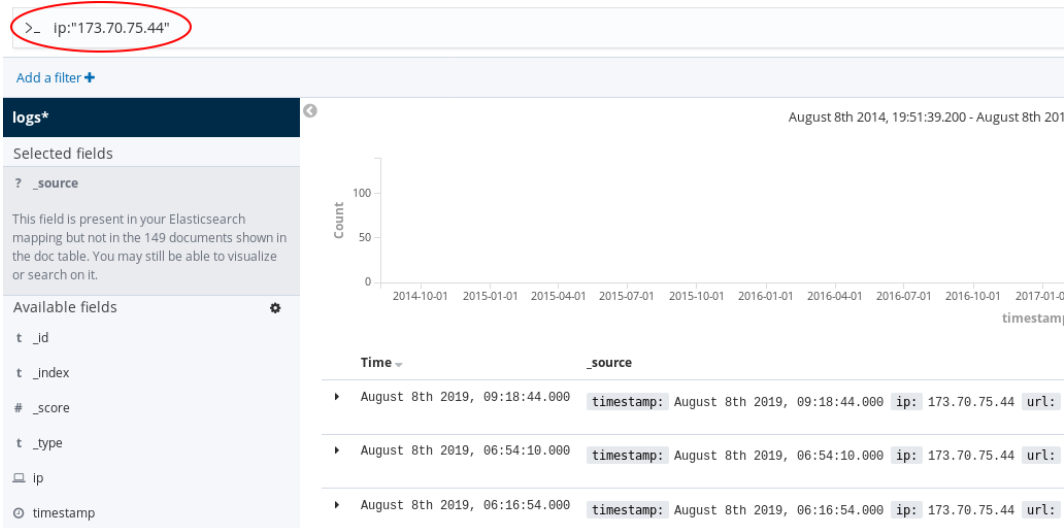
August 8th 2014, 19:47:35.000 - August 8th 2019, 19:47:35.000 — Auto

Time ▾

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

In this view, you can use the Lucene²⁸ query syntax, which is noticeably easier than the Elasticsearch DSL²⁹. For example, you can find requests with a specified IP using the following query:

```
ip:"173.70.75.44"
```



The screenshot shows a search interface with a query input field containing `ip:"173.70.75.44"`. Below the query is a sidebar with a list of fields under the heading "logs*". The main area displays a table of search results for the query. The table has columns for "Time" and "_source".

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

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

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

Alternatively, even a time range:

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

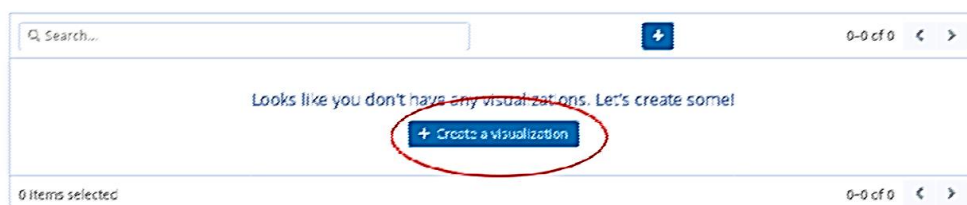
4.5.3 Exercise: find interesting data in the cluster.

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.

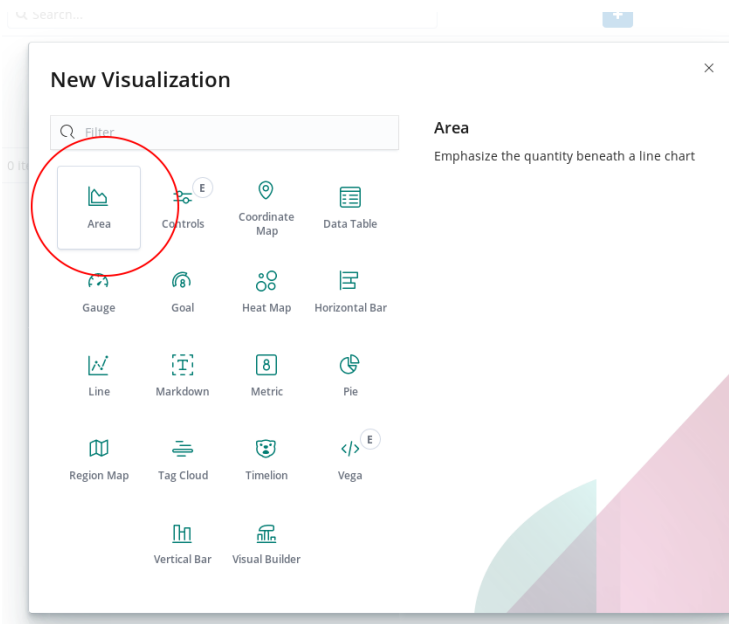


The screenshot shows a search interface with a search bar and a button labeled "+ Create a visualization". The button is circled in red. Below the search bar, there is a message: "Looks like you don't have any visualizations. Let's create some!".

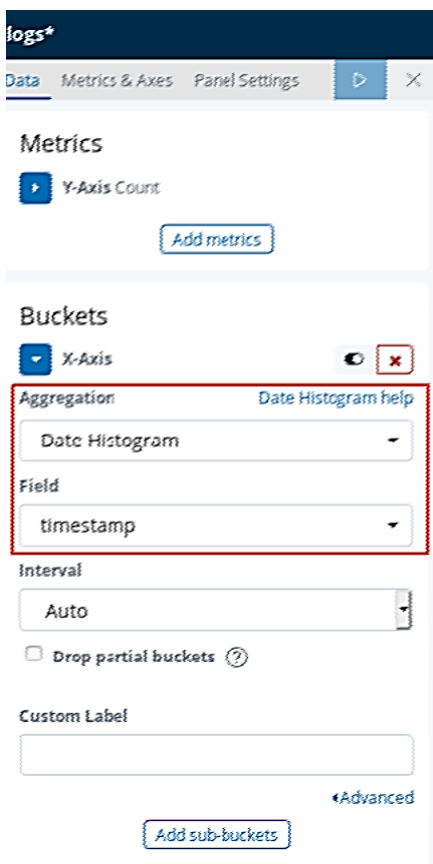
²⁸ <https://lucene.apache.org/>

²⁹ https://elasticsearch-dsl.readthedocs.io/en/latest/search_dsl.html

Select an area chart:



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



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 Set To Now To Set To Now

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

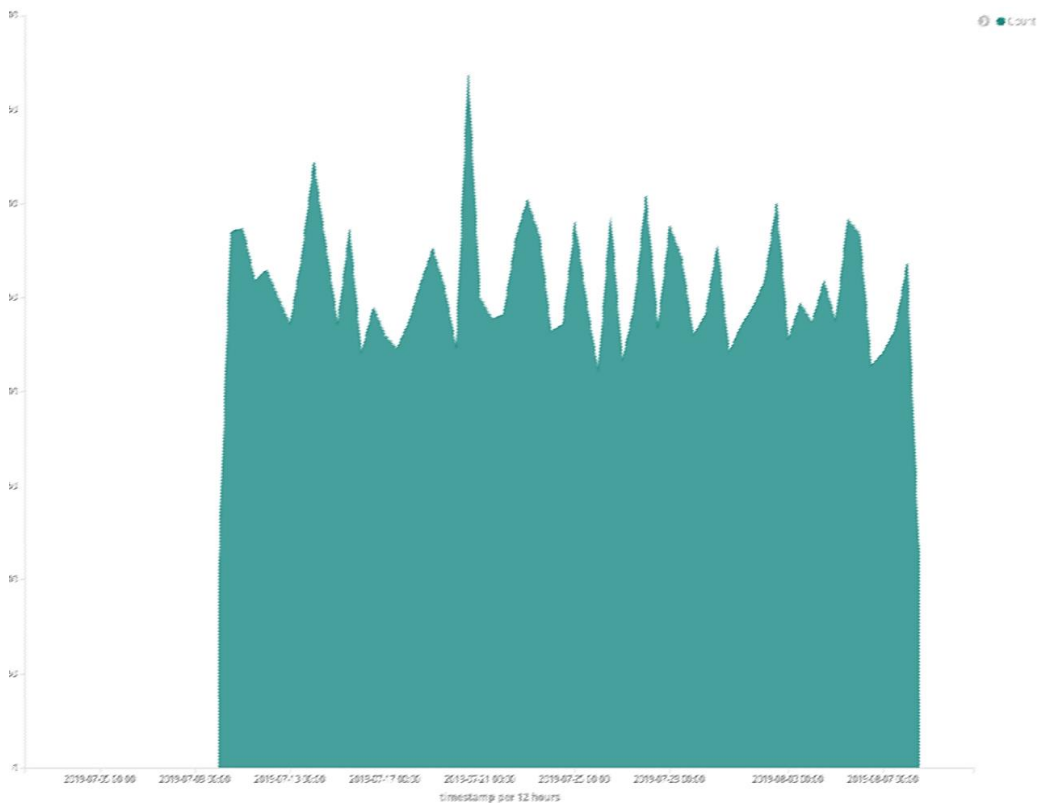
July 2019 August 2019

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 Logstash³⁰) 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 - SNARE³¹ in our case.

³⁰ <https://www.elastic.co/products/logstash>

³¹ <https://github.com/mushorg/snare>

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

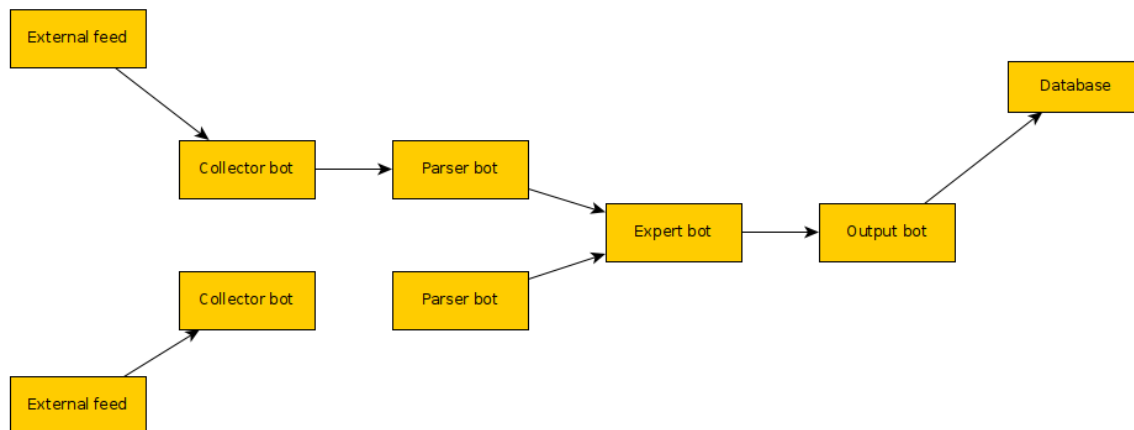
The whole idea of IntelMQ is based on so-called **bot nodes** and the connections between them.

Technically speaking, the bots are mere python scripts running on a single machine and communicating with one another using a Redis³² 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 Shodan³³ 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.ch³⁴) and outputs the gathered data to a local file without any additional processing.

³² <https://redis.io/>

³³ <https://www.shodan.io/>

³⁴ <https://abuse.ch/>

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
```

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/"/>	
postfix	<input type="text" value="ipblocklist.csv"/>	

- 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 'Yopt/shared/fpblocklist.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 'Yopt/shared/fpblocklist.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.identifi er"]
```

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

runtime	runtime	
column_regex_search	<input type="text" value="{}"/>	
columns	<input data-bbox="758 1534 1225 1590" type="text" value='["time.source", "destination.ip", "destination.port", "extra.l'/>	
default_url_protocol	<input type="text" value="http://"/>	
delimiter	<input data-bbox="766 1668 1212 1713" 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.

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
```

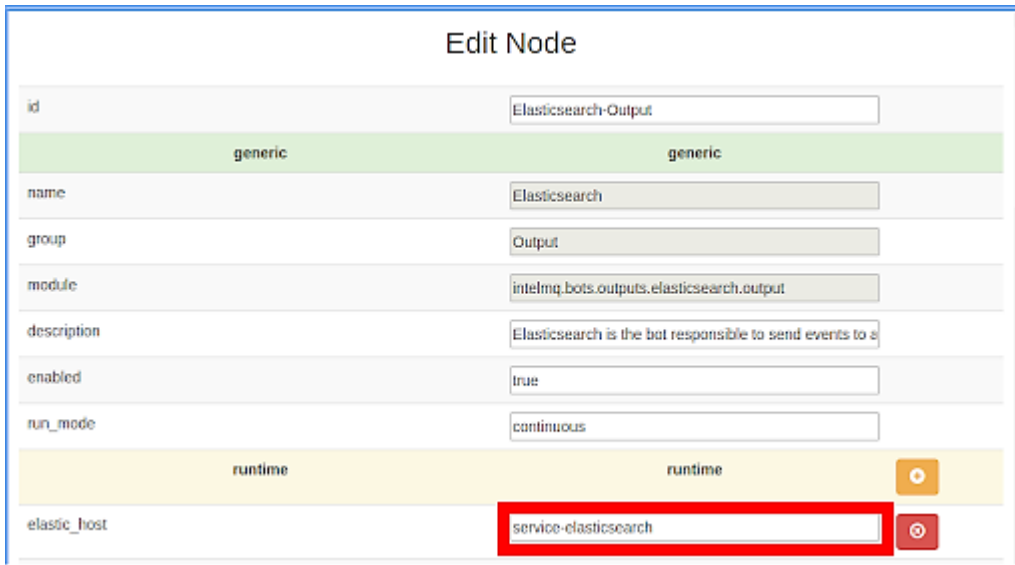
You can read more about creating custom bots here:

<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"`.



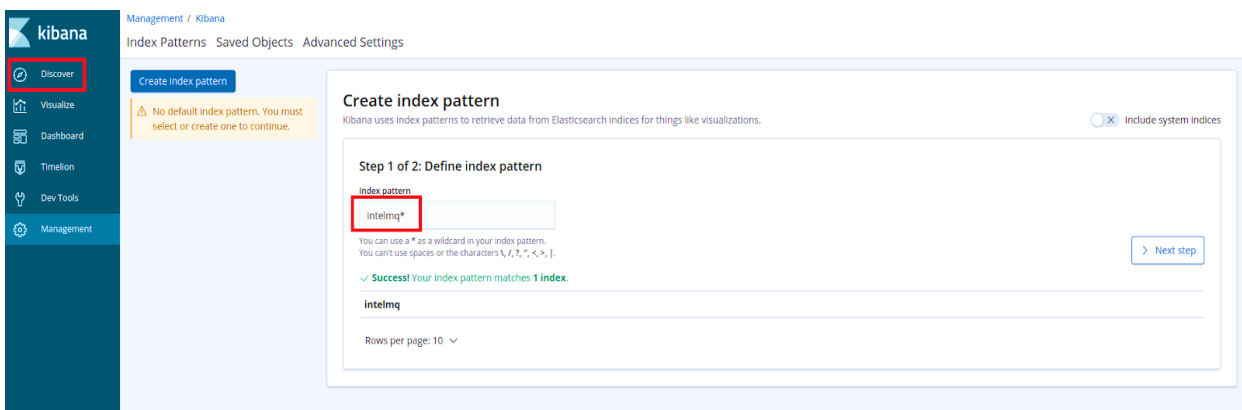
generic	
id	Elasticsearch-Output
name	Elasticsearch
group	Output
module	intelmq.bots.outputs.elasticsearch.output
description	Elasticsearch is the bot responsible to send events to e
enabled	true
run_mode	continuous

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

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  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: 97.201.206.232 raw: eyJzb3VyY2UudXJsIjogImh0dHA6Ly9ob25leXBvdC9pbnRleCIsICJ0aW11LnNvdXJjZSI6ICIyMDE5LTASLTl1V
00CswMdownMCIscjZb3VyY2UuaXAiOiAiMzcuMjAxLjIwN14yMzIiLCAiZXh0cmEuc6FyYw1zIjoge319 feed.url: file://localhost/opt/shared/snare_log.json feed.accuracy: 1
00 _id: IY0mhG4BhV5jksuX-Z60 _type: events _index: intelmq _score: -

┆ November 19th 2019, 18:12:45.000  source.url: http://honeypot/index time.source: September 25th 2019, 15:30:36.000 time.observation: November 19th 2019, 18:12:45.000 feed.name: __FEED__
feed.provider: __PROVIDER__ source.ip: 107.20.64.137 raw: eyJzb3VyY2UudXJsIjogImh0dHA6Ly9ob25leXBvdC9pbnRleCIsICJ0aW11LnNvdXJjZSI6ICIyMDE5LTASLTl1V
DEzOjMwOjM2KzAwOjAwIiwgInNvdXJjZS5pcCI6ICIxODUuNTUuMjYyLjE5NCIsICJ1eHRyYSwYXJhbXMiOiB7ImxvZ2luIjogIiBcIiBvc1AxPTEtLSIsICJzdWJtaXQ1OiAiIFN1Ym1pdCIscjZb3VyY2UuaXAiOiAiMzcuMjAxLjIwN14yMzIiLCAiZXh0cmEuc6FyYw1zIjoge319 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: I40mhG4BhV51ksuX-dEG tvpe: events _index: intelmq _score: -

┆ November 19th 2019, 18:12:45.000  source.url: http://honeypot/index time.source: September 25th 2019, 15:30:39.000 time.observation: November 19th 2019, 18:12:45.000 feed.name: __FEED__
feed.provider: __PROVIDER__ source.ip: 185.51.242.194 raw: eyJzb3VyY2UudXJsIjogImh0dHA6Ly9ob25leXBvdC9pbnRleCIsICJ0aW11LnNvdXJjZSI6ICIyMDE5LTASLTl1V
DEzOjMwOjM2KzAwOjAwIiwgInNvdXJjZS5pcCI6ICIxODUuNTUuMjYyLjE5NCIsICJ1eHRyYSwYXJhbXMiOiB7ImxvZ2luIjogIiBcIiBvc1AxPTEtLSIsICJzdWJtaXQ1OiAiIFN1Ym1pdCIscjZb3VyY2UuaXAiOiAiMzcuMjAxLjIwN14yMzIiLCAiZXh0cmEuc6FyYw1zIjoge319 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: JI0mhG4BhV51ksuX-dEx tvpe: events _index: intelmq _score: -

```

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 TheHive³⁵ – 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 MISP³⁶.

TheHive uses Elasticsearch as its database. In the training environment, the Elasticsearch instance used by TheHive is storing its files on another Kubernetes³⁷ container. Such a setup allows restarting TheHive container without losing data (that normally happens to all changes that were made inside the container).

Cortex³⁸ 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):

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

Followed by:

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

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

³⁵ <https://thehive-project.org>

³⁶ <https://www.misp-project.org>

³⁷ <https://kubernetes.io>

³⁸ <https://github.com/TheHive-Project/CortexDocs>

```
==> 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.ex
- 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 GeolIP³⁹:

- 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.

³⁹ <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.

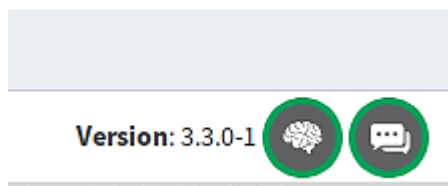
6.2.4 Configure the Hive-MISP integration and check if alerts are fetched

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.



7. ANALYSTS PART - GENERAL INFORMATION

7.1 INTRODUCTION

This part covers the training modules aimed towards the security analysts that will be using the tools to assist them in their duties and investigations up.

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 VirtualBox⁴⁰ in the main operating system of the computer,
- 40 GB of free disk space (SSD recommended).

7.2 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	-	-

⁴⁰ Oracle VirtualBox virtualisation software can be downloaded for free from this website: <https://www.virtualbox.org/>

8. MISP ANALYST

8.1 INTRODUCTION

Parameter	Description	Duration
Main Objective	MISP analyst workshop introduces trainees to basic MISP usage concepts. The concepts that will be described include creating and publishing events, adding attributes, searching through events, intel correlation, usage of galaxies and taxonomies.	-
Targeted Audience	Exercise is dedicated to members of SOC/CERT/CSIRT teams.	
Total Duration	1,5 hours	90 minutes
Time Schedule	Introduction to the exercise	10 minutes
	Events	60 minutes
	Galaxies	10 minutes
	Taxonomies	15 minutes

This exercise is designed for the analysts willing to expand their knowledge about MISP use cases and overall usage.

Credentials to the Virtual Machine (VM): **enisa:enisa**

8.2 PRECONFIGURED STATES

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

8.2.1 misp-bare

This state consists of two MISP systems. One (MISP1: <https://misp.enisa.ex>) is not configured at all. This represents the state after admin configuration.

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

Another instance (MISP2: <https://misp2.enisa.ex>) contains data and minimal configuration. Credentials: admin@admin.test:SecondInstancePassword123!

For easier use of both instances, you can run two browsers in the VM (at least one in incognito mode) and then login to both MISPs at the same time.

8.2.2 misp-configured

This represents both MISP instances in the state after the exercise is finished. Follow the steps below to get to this stage from the misp-bare snapshot state.

8.3 PREPARATION

Now we will prepare the exercise environment on the Virtual Machine (VM).

8.3.1 Environment setup

To enable the exercise that contains the two MISP instances, navigate to the following folder using the console in the VM:

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

In that folder, type the following commands

```
./reset_data.sh and then ./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.

8.3.2 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
```

8.3.3 Login in MISP1

Log into your organisation's MISP instance with a web browser (<https://misp.enisa.ex>). This instance is only accessible from within the provided VM.

We will start by explaining what events are and what you can do with them.

8.3.4 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.

A quick overview of the events view is presented in the image, taken from the MISP book <https://www.circl.lu/doc/misp/>.

Figure 3: Events view from the MISP book

A. Add Event

1. Add Event

2. Populate Fields

3. Choose File

4. Add

B. Add Attachments

7. Add Attachment

9. Upload

C. Add Event Attributes

5. Populate Fields

6. Submit

The following attribute types should be added for each event:

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

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

A detailed description on how to add an event is described below.

8.3.5 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 that adds 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 (IoC's)*, *Command & Control Server (C&C) addresses*, *md5 hashes*, or other additional information. Many types of attributes exist

8.4 EXERCISES

8.4.1 Exercise 1 - Adding an Event

Imagine that you have observed a new malware sample inside your organisation. It was not able to infect any of the hosts inside your organisation but you collected the sample and started the analysis.

Information about the sample can be found in the REPORT.pdf file.

Read the report and try to add an event to your MISP instance that describes all of the information contained in the report. Make use of the correct tags, taxonomies and attributes.

If you are stuck, below are the exact instructions of how it can be done:

- Go to Event Actions -> Add Event, and fill the fields appropriately.
- The Date, Event info and Analysis fields are obvious.
- We set threat level and distribution according to the descriptions presented in the previous chapter. (8.3.5).

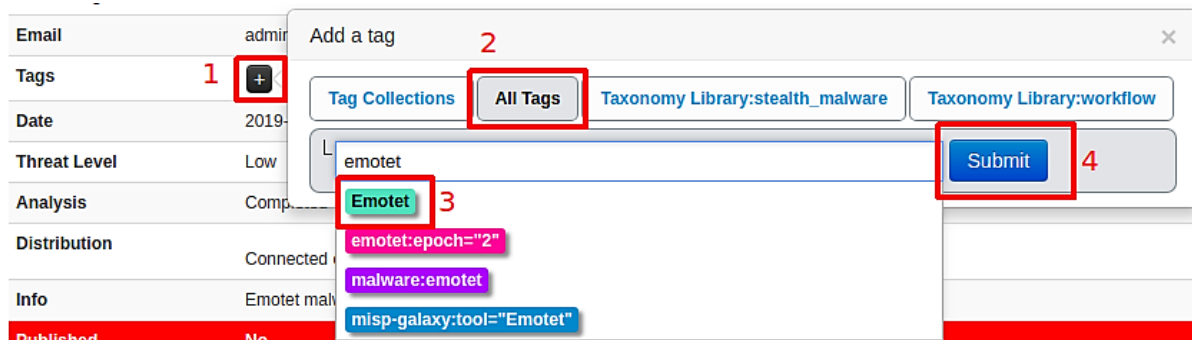
Add Event

Date	Distribution ⓘ
<input type="text" value="2019-10-02"/>	<input type="text" value="Connected communities"/>
Threat Level ⓘ	Analysis ⓘ
<input type="text" value="Low"/>	<input type="text" value="Completed"/>
Event Info	
<input type="text" value="Emotet malware campaign IoCs"/>	
Extends event	
<input type="text" value="Event UUID or ID. Leave blank if not applicable."/>	
<input type="button" value="Add"/>	

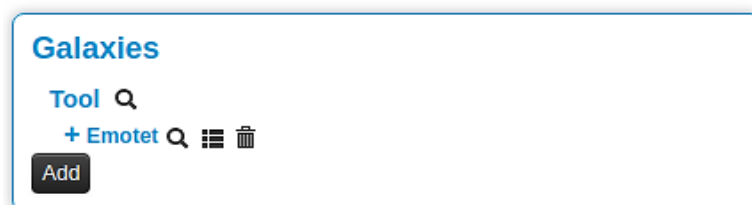
For easier grouping and correlation, we should add a tag describing the malware family; we can just put *emotet* inside the Add a tag form.

There are multiple options to choose from and ideally, we should add all of those that describe it as *emotet*.

Another important tag to set is the TLP tag, it describes sharing permissions of the intel according to the Traffic Light Protocol⁴¹.



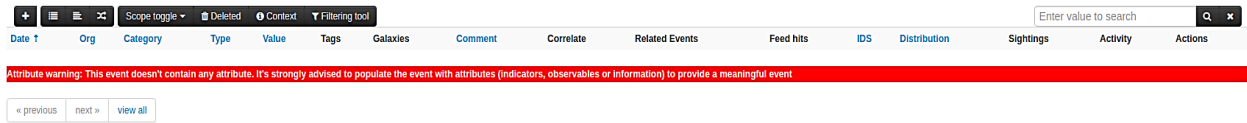
If you choose the `misp-galaxy:tool="Emotet"` and click submit you can observe that no tag was added. What actually happened is that you have added the event to the *emotet* galaxy. You can observe that further down on the screen.



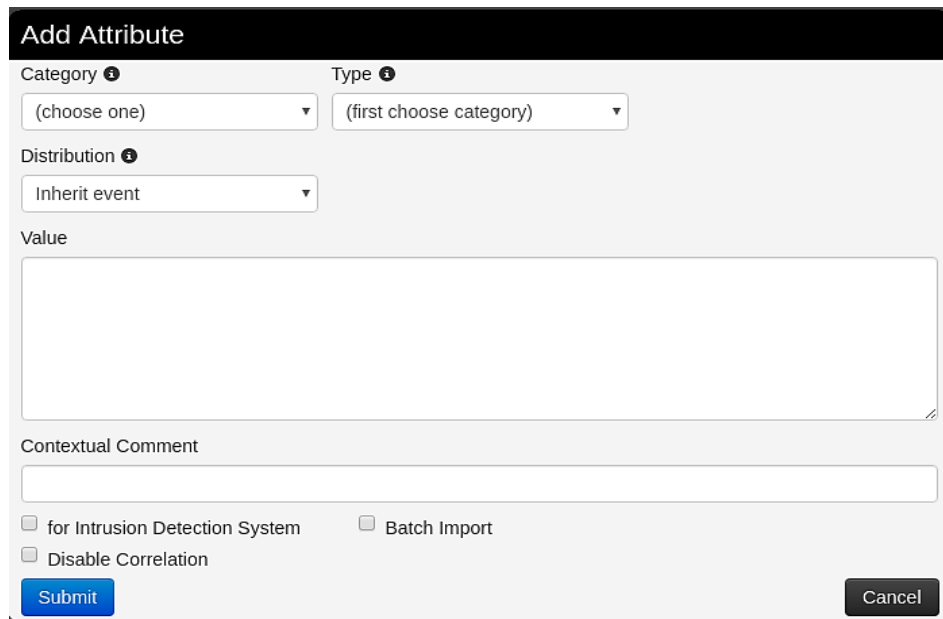
More details about galaxies can be found in the next sections.

⁴¹ <https://www.first.org/ttp/>

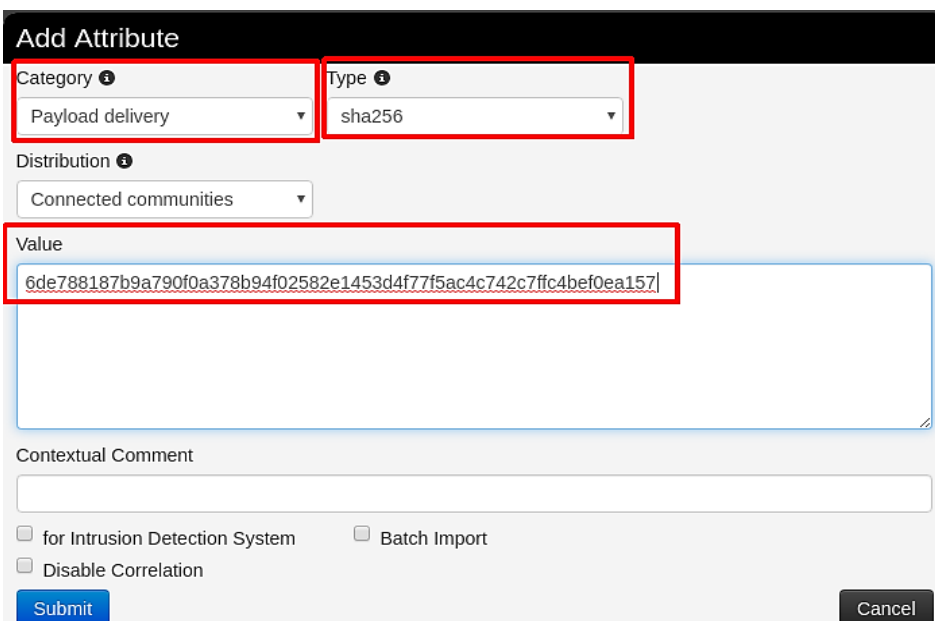
Last of the important things to do is to add some attributes describing the event. The Attributes section is located at the bottom of the page.



To add a new attribute, you should click the + sign in the top left corner. You will be presented with the following form:



For example, to add the sha256 checksum of the sample, you should fill the fields of the form in a similar way as presented in the image below:



Now try adding the rest of the attributes on your own.

This should result in the attributes section looking like this (your result may vary):

Date	Org	Category	Type	Value	Tags	Galaxies	Comment	Correlate	Related Events	Feed hits	IDS	Distribution	Sightings	Activity	Actions
2019-10-16		Network activity	ip-dst	104.18.60.46	+	Add		☑				Inherit	👁️👁️👁️ (000%)		🗑️ 📄
2019-10-16		Network activity	ip-dst	104.199.245.51	+	Add		☑				Inherit	👁️👁️👁️ (000%)		🗑️ 📄
2019-10-16		Network activity	ip-dst	66.228.39.137	+	Add		☑				Inherit	👁️👁️👁️ (000%)		🗑️ 📄
2019-10-16		Network activity	ip-dst	108.58.41.242	+	Add		☑				Inherit	👁️👁️👁️ (000%)		🗑️ 📄
2019-10-16		Network activity	ip-dst	47.100.43.55	+	Add		☑				Inherit	👁️👁️👁️ (000%)		🗑️ 📄
2019-10-16		Payload delivery	url	https://sodadino.com/wp-admin/gczk/	+	Add		☑				Inherit	👁️👁️👁️ (000%)		🗑️ 📄
2019-10-16		Payload delivery	url	http://newgensolutions.net/joomla_30n0k0/	+	Add		☑				Inherit	👁️👁️👁️ (000%)		🗑️ 📄
2019-10-16		Payload delivery	url	https://mokhoafacebookkn.com/wp-content/themes/lalita/Kj6VM3sio/	+	Add		☑				Inherit	👁️👁️👁️ (000%)		🗑️ 📄
2019-10-16		Payload delivery	sha256	6de788187b9a790f0a378b94f02582e1453d4f77f5ac4c742c7ffc4bef0ea157	+	Add		☑				Connected	👁️👁️👁️ (000%)		🗑️ 📄

As you can see, attributes have tags and galaxies sections as well, this allows for better granularity in describing the event and payloads associated with it.

Click around and explore. Try to find tags or galaxies useful for attributes you have just added.

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

Now let us see how the event presents itself on the events list.

- Event Actions -> List Events

8.4.2 Exercise 2 - Search and Correlation

In this exercise, we will focus on search and filtering abilities of MISP. The experience gained from working with events obtained in previous the exercise will come in handy.

Try to find all unclassified events in MISP that may be correlated in any way with the event you added in the previous exercise. There are multiple ways to correlate in MISP. Through search, Correlation Graph, Related Events view or in the Attributes view.

What related events were you able to find?

If you are stuck, below are some tips of how it could be done.

Search can be found in the Event List view, accessible by Event Actions -> List Events.

NOTE: all events on the list are actually from different organisation than ours. Our organisation is called **MY-SUPER-CERT** and the feed inside the MISP instance is from organisation **ORGNAME**.

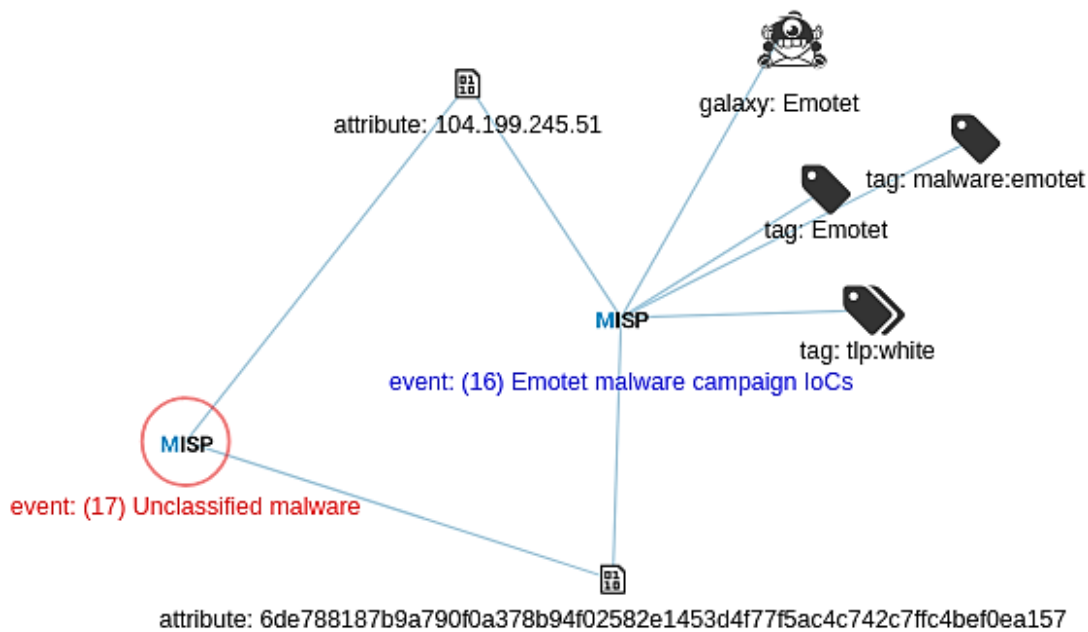
- In the Search field you can enter any value from the attribute and all the associated events should be presented.
- Try `6de788187b9a790f0a378b94f02582e1453d4f77f5ac4c742c7ffc4bef0ea157`
- Or `104.199.245.51`.

- Keep in mind that this can sometimes take a little bit longer (for large datasets). See what other related events you can find with this method.

View Correlation Graph can be found in event view on the left side.

In this case, we can easily see an event that shares two attributes with the one we created. You can expand nodes by selecting them and pressing Ctrl+x.

This powerful tool enables you to investigate whole clusters of malware. Try to click around and see what else you can find.



Related Events can be found in event view on the right side on the top. Without performing any actions, you can see here potentially related events based on attributes.

Related Events

2019-10-02 (16)

Orgc: [MY-SUPER-CERT](#)

Date: [2019-10-02](#)

Info: [Emotet malware campaign IoCs](#)

By searching and correlating you can clearly see how powerful and important attributes entered into MISP instances are.

8.4.3 Galaxies

You previously already encountered a galaxy; below you can find a description on how they work.

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.

To add a galaxy to the event go to the detailed event view of the event you created in the previous chapter. Then scroll down to Galaxies and click on Add. Choose All namespaces, in Select an Option and select an appropriate malware type and family, then click Submit.

8.4.3.1 Examples of galaxies

Here we list some examples of potentially interesting galaxies:

- **Ransomware:** galaxy with information on ransomware campaigns and families, based on <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.

8.4.4 Taxonomies

You already used a taxonomy before: TLP, below you can find description on how they work.

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.

As with galaxies, we can try them out in our event we created earlier. Find your event in List Events view once more.

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

8.4.4.1 Popular taxonomies

- **TLP (Traffic Light Protocol):** classification of sensitive information distribution. There are 4 TLP levels⁴²:
 - **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>

⁴² <https://www.first.org/ttp/>

9. LOGS ANALYSIS ANALYST

9.1 INTRODUCTION

Parameter	Description	Duration
Main Objective	This exercise will present advanced tools for analysts that simplify log management and incident handling. The tools - most notably, IntelMQ and Elasticsearch - will be introduced at the beginning, and then they will be tested in a scenario closely resembling a real world exploit hunting case.	-
Targeted Audience	The exercise is dedicated to CSIRT/SOC staff responsible for monitoring and incident analysis	
Total Duration	2 hours	120 minutes
Time Schedule	Introduction to the exercise	30 minutes
	IntelMQ getting started and initial configuration	30 minutes
	Exercises: hunting exploits using available tools	60 minutes

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

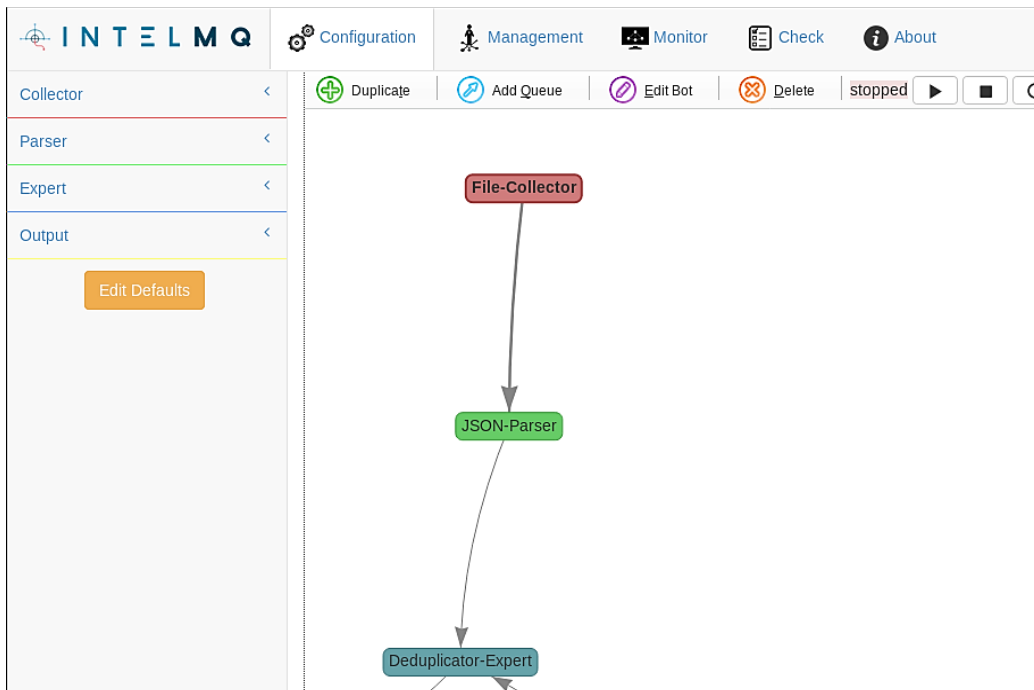
Trainees will have a chance to learn about a wide range of well-known software that is very useful during malware analysis, and to see how a pre-defined combination of common tools works in practice.

9.2 OVERVIEW OF INTELmq

INTELmq

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

IntelMQ's main strength is the backend queue, but it also features a dedicated web-UI that greatly simplifies configuration management.



IntelMQ has 4 types of entities:

- **Collectors**, that produce messages and pass them further into the system.
- **Parsers**, that convert unstructured data into structured messages. For example, you can use them to parse publicly available feeds into a format that IntelMQ will understand.
- **Experts**, that operate on parsed data and enrich or change it. For example deduplicator, or revdns experts.
- **Outputs**, that send parsed data to other systems.

In this exercise, we are using five different bot-types:

- **File Collector**: collector that cyclically reads data from a file on the disk and passes the data into the system
- **JSON-Parser**: parser that reads JSON-serialised messages from input and converts them into a structured format understood by IntelMQ. This allows other bots to “understand” the meaning of JSON fields. For example: RevDNS expert needs to know which field corresponds to IP to do its work. Assigning such meaning to fields is a job for a parser.
- **Abusech-IP-Parser**: another parser but this one was created for a specific feed - AbusechIP⁴³.
- **Deduplicator-Expert**: with multiple feeds as input sources, duplicated events can become a problem. In our exercise scenario, we only have two data sources, but in real world situations, one often works with dozens of feeds. Deduplicator keeps events in a temporary database for a configurable amount of time, and drops already seen events.
- **Elasticsearch-Output**: quite straightforward: it stores processed events in a configured Elasticsearch database.

How can it be useful for analysts? A very common use case for a SOC or CSIRT is the monitoring of multiple feeds, and reacting to them.

⁴³ <https://abuse.ch/>

In many cases, the actions can become a bit repetitive. For example, imagine multiple external feeds of malicious domains and IPs (IntelMQ supports many such data sources). Common operations include domain resolving, geo-IP lookup, filtering based on the geo-IP country or TLD, finally de-duplication, and reporting by saving output to a file, email, webhook or database.

All these actions and much more can be automated by IntelMQ. Time saved thanks to this will quickly dominate the initial time investment spent for configuring and testing the pipeline.

In this exercise, we will look at one such pipeline and follow the actions of a hypothetical analyst.

9.3 CONFIGURE THE EXERCISE

9.3.1 Ensure that DNS is configured properly

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

```
$ dig -ta +short intelmq.enisa.ex
127.0.0.1 # or any other valid IPv4
$ dig -ta +short kibana.enisa.ex
127.0.0.1 # or any other valid IPv4
```

9.3.2 Apply the helm configuration file

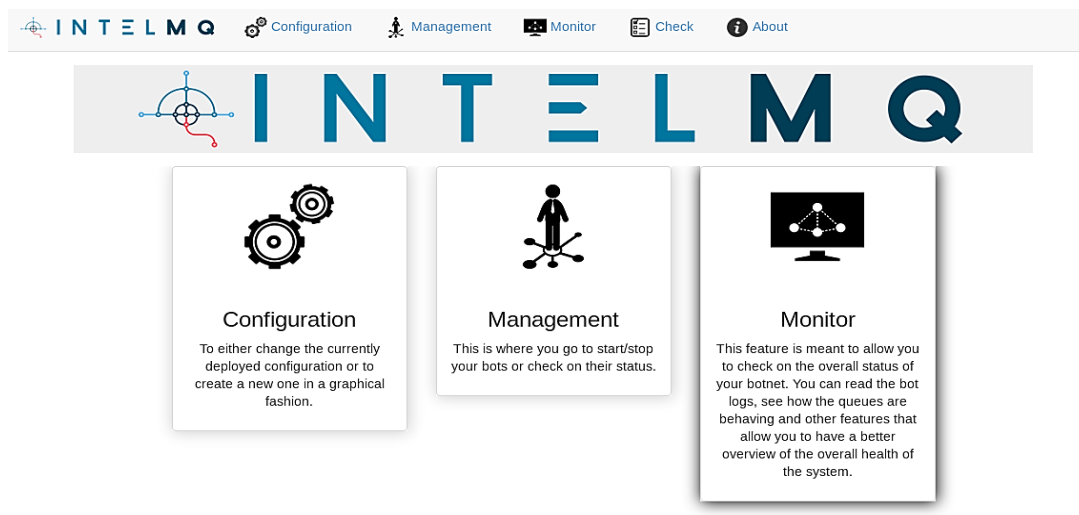
```
cd /opt/enisa/trainings-2019/analyst/intelmq/
$ helm install intelmq/
```

9.3.3 Completion of the installation

It can take up to a few minutes before all the tools are downloaded and ready.

9.3.4 Ensure that Elasticsearch works correctly

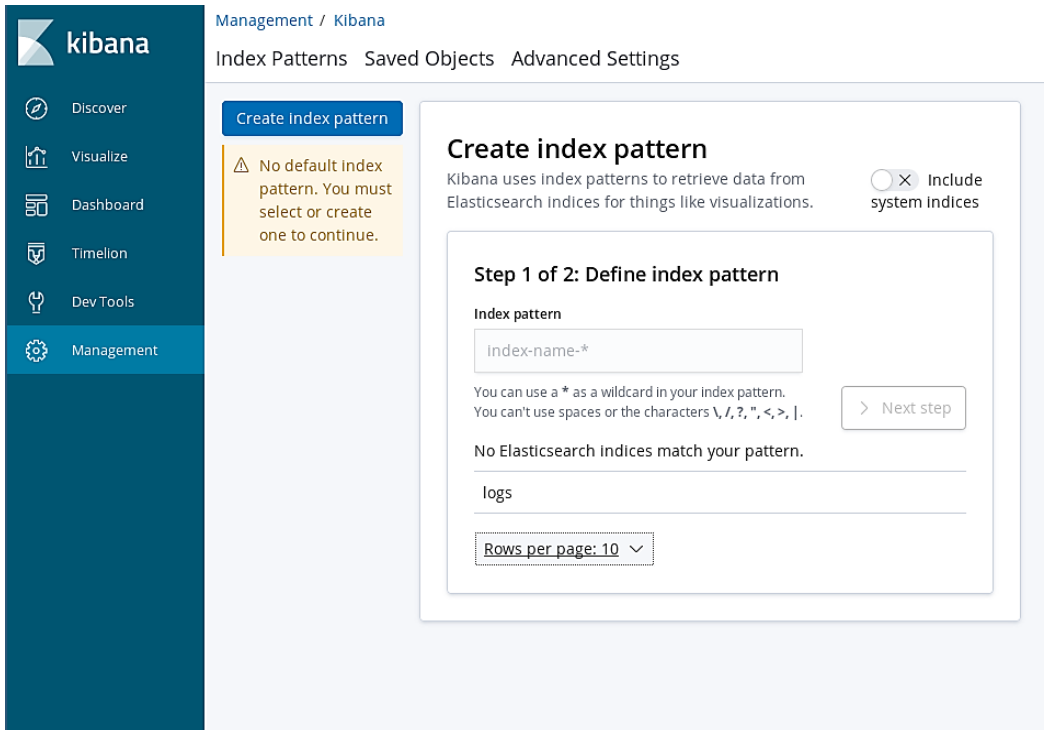
Point your browser to <http://intelmq.enisa.ex>. You should see the following:



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

9.3.5 Ensure that Kibana works correctly.

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



Or use the command line:

```
$ 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
<
```

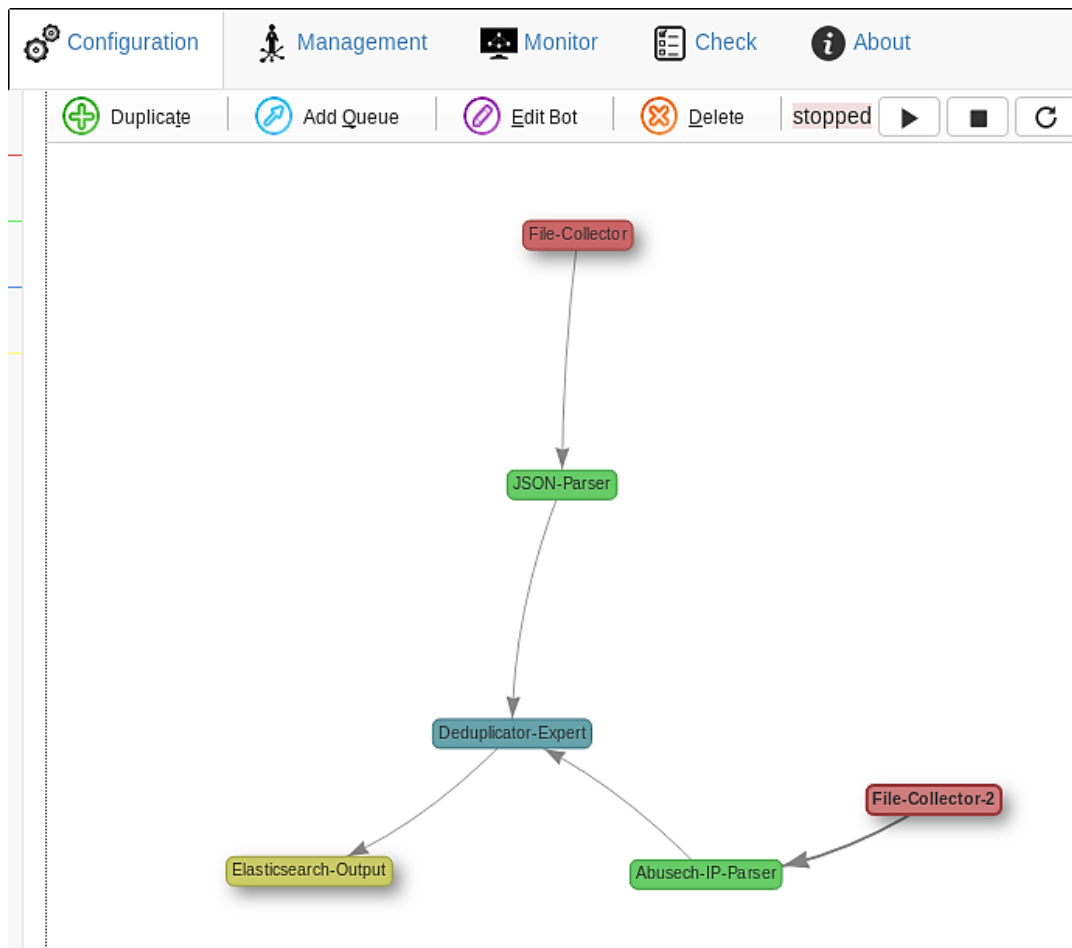
HTTP 302 Found means that everything is working correctly.

9.4 GET FAMILIAR WITH INTELmq

In this exercise, we will create a simple IntelMQ pipeline. We will retrieve data from our local simulated honeypots, from freely available IP blacklists, de-duplicate the results and save them all to the Elasticsearch database.

9.4.1 Get familiar with the pipeline

Look at the pipeline configured in your IntelMQ instance:



There are two collectors: File-Collector and File-Collector-2.

File-Collector is the first one. It reads the `http_logs.json` file. You can preview that file by opening `/opt/enisa/trainings-2019/analyst/intelmq/shared/http_logs.json` with your favourite text editor. After reading it, this file is parsed by the JSON-Parser to the structured format understood by IntelMQ.

The second collector is File-Collector-2. It reads a `blacklist.txt` file (you can find it in the same directory). After reading it, this data is parsed by a dedicated parser (Abusech-IP-Parser).

The Deduplicator-Expert de-duplicates all these sources, and the de-duplicated results go straight to the Elasticsearch-Output. Deduplicator has a temporary database where it keeps all events it has seen for a configurable amount of time (a common setting is 24h or 48h). When the same event goes through deduplicator multiple times, all but the first occurrences are dropped. This helps to reduce noise if we are reporting results of the pipeline to external organisations.

9.4.2 Start the botnet

By default, the pipeline is stopped. You can check its status by going to the management tab:

The screenshot shows the 'Management' tab of the INTEL MQ interface. On the left, there are three status panels: 'Whole Botnet Status', 'Collectors Status', and 'Parsers Status', all showing a status of 'stopped'. On the right, the 'Individual Bot Status' panel shows a table of bot components, all with a status of 'stopped'.

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

Start the IntelMQ botnet by clicking the play button on the left. If everything goes fine, the result should look like this:

The screenshot shows the 'Management' tab after the botnet has been started. The status panels on the left now show 'running'. The 'Individual Bot Status' table on the right shows all bot components with a status of 'running'.

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

You can inspect a bot's status by clicking on it. For example, you can read its logs and ensure that there are no unexpected errors:

The screenshot shows the 'Logs' view for the 'JSON-Parser' bot. It displays a list of log entries with columns for Time, ID, Level, and Message. The log level is set to 'All' and the page shows 10 records per page.

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

9.4.3 Familiarise yourself with the honeypot

There is a honeypot running in your network. You can visit it by opening the following URL: <http://honeypot.enisa.ex/> in your browser.

This honeypot is powered by the Snare project. Snare is the successor of the Glastopf project⁴⁴. It is a scalable web application honeypot, attracting malicious agents and logging the interesting events. It is not doing any analysis - this job is forwarded to the Tanner. Tanner's job is to evaluate Snare events, serve dorks, and to adopt and change responses, to maximise attack surface^{45,46}.

Visit <http://honeypot.enisa.ex/> now. You should see the following empty-looking website:

Example Domain

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

[More information...](#)

Refresh the webpage a few times.

Now take a look at the `/opt/enisa/trainings-2019/analyst/intelmq/shared/snare.log` file. You should see logs similar to the following:

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

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

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

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

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

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

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

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

This means that SNARE is working and collecting logs correctly.

Unfortunately, there is no built-in support for SNARE logs in IntelMQ (adding new bot types is beyond the scope of this exercise).

⁴⁴ <http://mushmush.org/>

⁴⁵ <https://snare.readthedocs.io/en/latest/index.html>

⁴⁶ <https://github.com/mushorg/snare>

We need to convert them to json format first. In order to do this, go to the /opt/enisa/trainings-2019/analyst/intelmq/shared directory and type:

- `python3 parse_logs.py snare.log snare_log.json`

If you take a look at the snare_log.json now, you will see the same data, but in the .json format. IntelMQ will automatically pick up this data, parse it and send it through the pipeline.

9.4.4 Take a look at the data

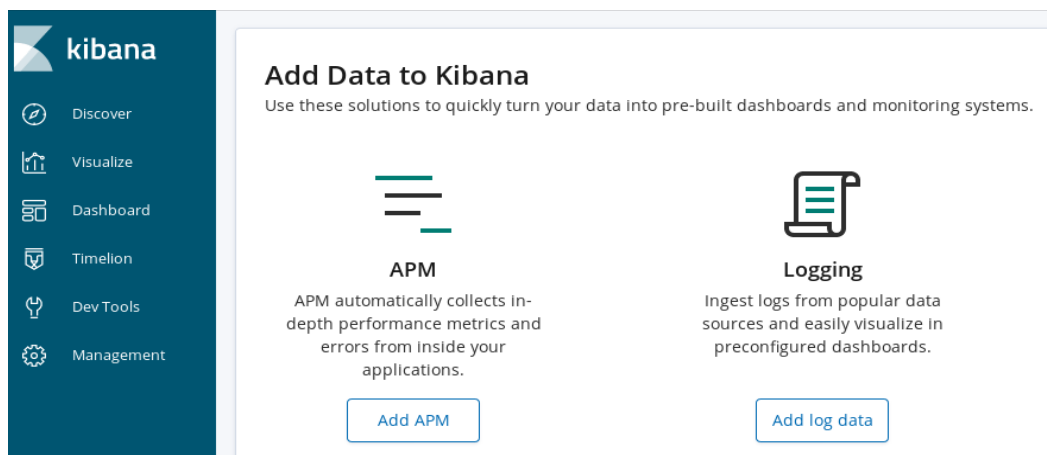
Elasticsearch is a very popular NoSQL database, used across many industries. It is fast, scalable, and it's main and original strength was fast search including full-text searches.

This speed does come at a cost though. Elasticsearch needs a lot of RAM, and its query language is quite limited. For example, aggregations, subexpressions and joins are not available directly (or are very limited).

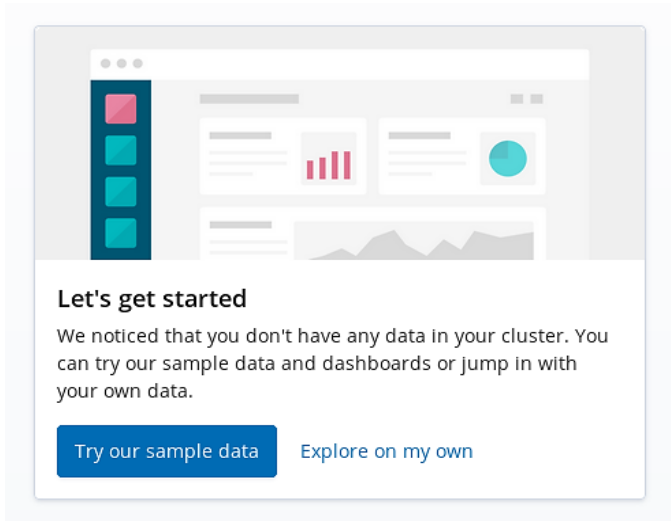
This is very different from SQL databases, whom allow programmers to write arbitrarily complex (but potentially slow) queries. Filtering on complex predicates in Elasticsearch is usually done by precomputing before inserting the data and adding results as additional fields.

Kibana is a web UI for the Elasticsearch database. It can be very useful for browsing and understanding the data you are dealing with.

First, open <http://kibana.enisa.ex> in your browser. You should see this form:



You may notice a generic let's get started message instead:



In this case, click the Explore on my own button. You might also want to ensure that you have started the IntelMQ pipeline, and that it is running.

Now click on the management tab and create an index pattern. In order to do this, enter IntelMQ as an index pattern name.

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 \, /, ?, ", <, >, |.

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

intelmq

Rows per page: 10 ▾

Then select time.observation as a time filter field, and then finally click Create index pattern:

Step 2 of 2: Configure settings

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

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](#)

You can browse the data in the Discover mode:

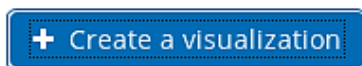
The screenshot shows the Kibana Discover interface. At the top, it displays '14,375 hits' and a search bar containing '> Search... (e.g. status:200 AND extension:PHP)'. The left sidebar contains navigation options: Discover, Visualize, Dashboard, Timelion, Dev Tools, and Management. The 'Discover' view is active, showing a list of selected fields for 'intelmq' and a list of available fields. The main area displays a bar chart titled 'October 9th 2018, 22:39:38.900 - October 9th 2019, 22:39:38.900' with a time range of 'Auto'. The chart shows the count of observations per week from May 2019 to September 2019. Below the chart, there is a table of search results with columns for 'Time' and '_source'. The results show three entries for September 20th 2019, each with a 'time.observation' field and a '_source' field containing log data.

Remember to change the time range in the upper right corner - the default is 15 minutes. Change it to something much longer, for example: 1 year.

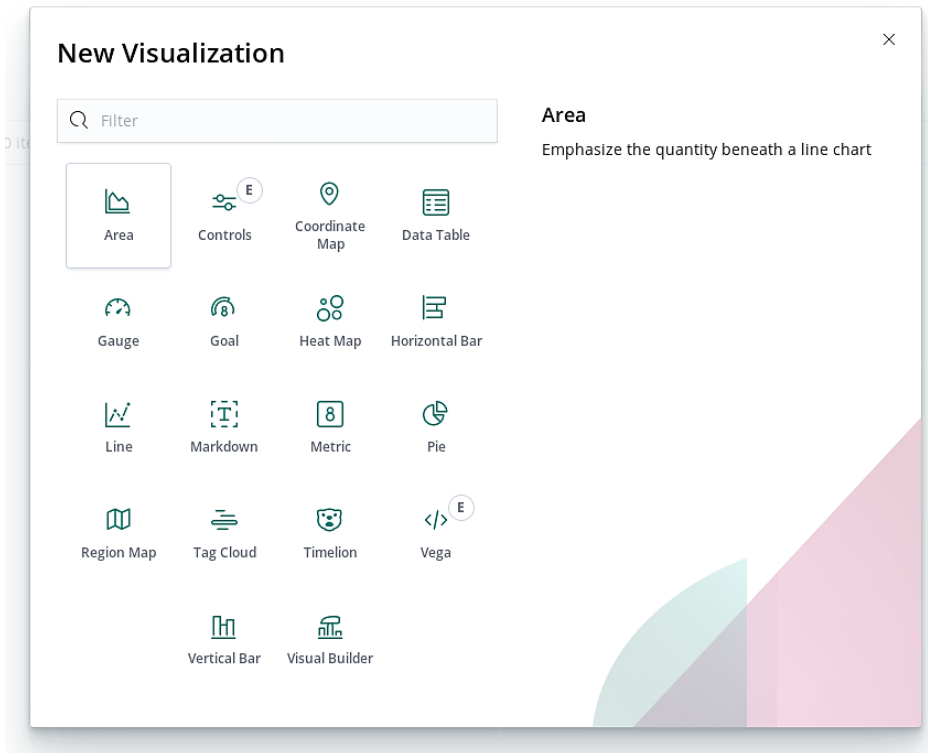
This screenshot shows the Kibana Discover interface with a different time range. The top navigation bar includes 'New', 'Save', 'Open', 'Share', 'Inspect', 'Auto-refresh', and 'Last year'. The search bar is empty. The time range is set to 'Auto'. The bar chart shows the count of observations per week from May 2019 to October 2019. The chart shows a significant peak in late May 2019, followed by a period of lower activity, and then a second peak in late September 2019.

The real strength of Kibana are its visualisations. Let us create a simple visualisation. First, select Visualise from the left:

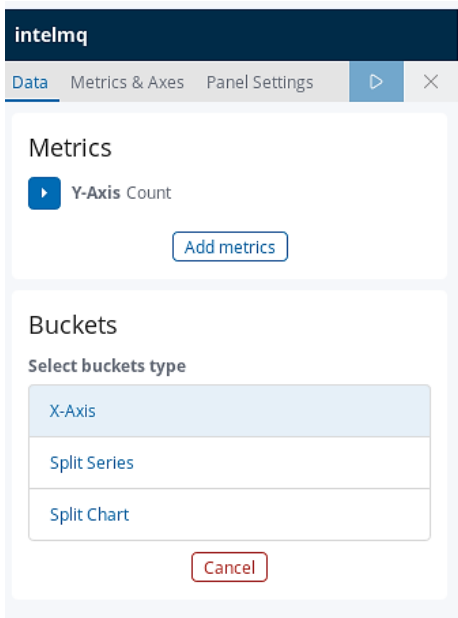
Looks like you don't have any visualizations. Let's create some!



And select area chart:



Then pick IntelMQ as an index (it is the only option) and add a bucket for the X-axis:



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

Buckets

X-Axis

Aggregation [Date Histogram help](#)

Date Histogram

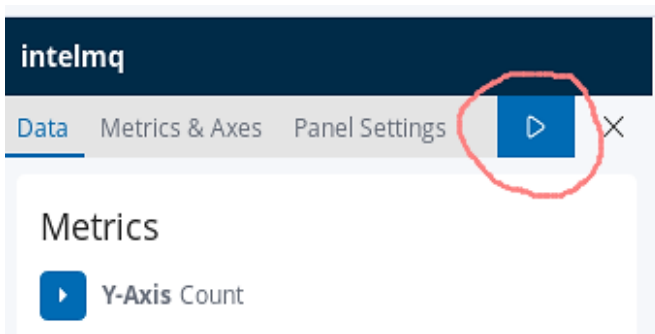
Field

time.observation

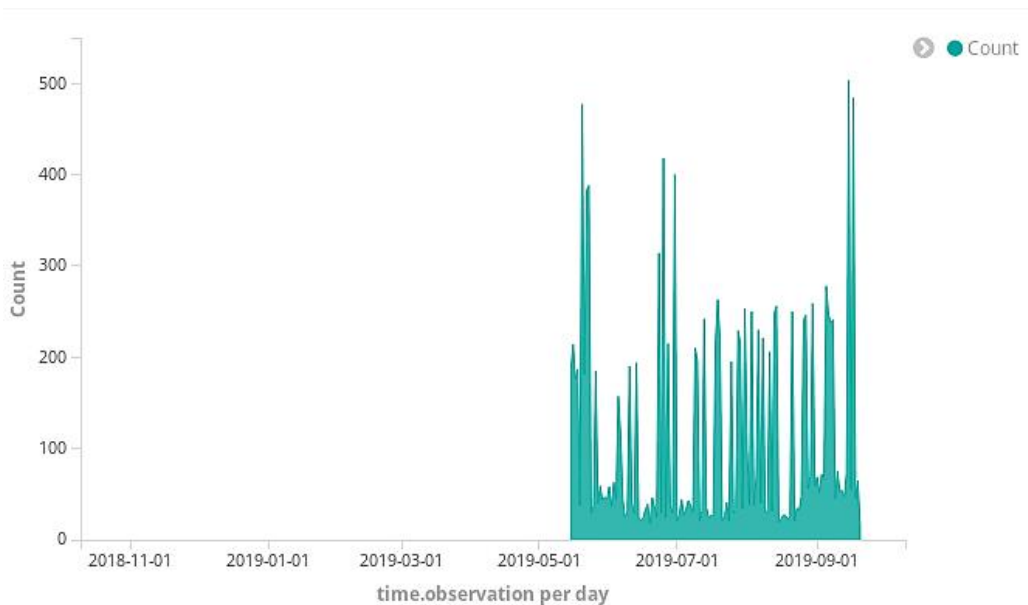
Interval

Daily

Confirm with the “play” button above:



You should see a graph similar to this one:



As you can see, request count distribution is not even. This means that we have more requests on some days than the others.

9.4.5 Exploit hunt

Let us use our instinct to find potential exploits in the indexed data.

For example, we can search for suspicious request paths. Let us use a simple Lucene⁴⁷ query in Kibana for that.

What is Lucene? It is a search engine software library originally written in Java. It is used in many search projects, most famously Apache Solr. However, Lucene is not only a library; its query syntax is quite simple, and allows operators to easily select the data they are interested in. Because of this, the Lucene query language became adapted by multiple software projects, including Elasticsearch and MWDB⁴⁸.

There are multiple ways to write a Lucene query:

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

More documentation can be found on the Elasticsearch website:

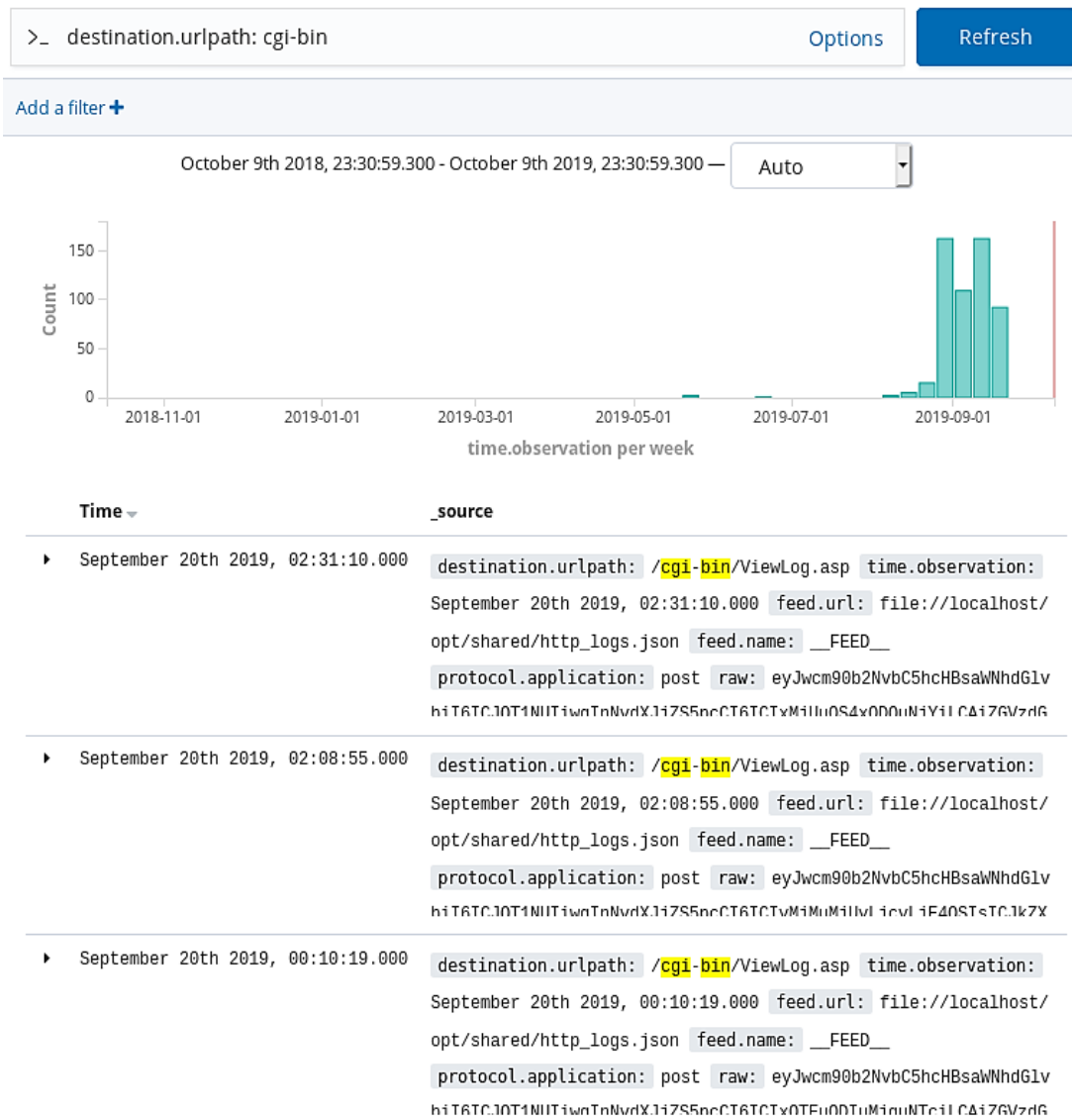
<https://www.elastic.co/guide/en/elasticsearch/reference/7.4/query-dsl-query-string-query.html#query-string-syntax>

Select Discover in the menu on the left, and type `destination.urlpath: "cgi-bin"` in the big search box on the top. This will allow us to find all URL paths with `cgi-bin` as a URL component. The result should look like the next page.

⁴⁷ <https://lucene.apache.org/>

⁴⁸ <https://www.cert.pl/en/news/single/mwdb-our-way-to-share-information-about-malicious-software/>





The /cgi-bin/ folder is a traditional location for CGI scripts. CGI is a very dated technology, one of the first methods used to create interactive websites. CGI scripts are often written in insecure languages and buggy, which makes them a common target of exploits. Our honeypot obviously has no CGI support, so we know that all the CGI requests are malicious and probably an exploit attempt.

9.4.6 Exercise 1

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

9.4.7 Exercise 2

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

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

Filter by requests that have some data submitted. Add a filter, this time using an UI.

First, click the “Add a filter” button:

45,305 hits

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

[Add a filter +](#)

intelmq*

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

[Add a filter +](#)

Add filter ✕

Filter Edit Query DSL

extra.params.su|

- intelmq*
- extra.params.submit**
- extra.params.submit.keyword

Cancel Save

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

[Add a filter +](#)

Add filter ✕

Filter Edit Query DSL

extra.params.submit

Label

Optional

Cancel Save

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

intelmq*

Selected fields

? _source

Available fields ⚙

t _id

t _index

_score

t _type

t extra.params.login

t extra.params.password

t extra.params.submit

The result should look like this:

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

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

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

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

9.4.8 Exercise 3

Now let us look at the comments. Remove the filters and selected fields.

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

Time	extra.params.comment
October 15th 2019, 23:41:21.000	<script>prompt(1)</script>@gmail.com<isindex formaction=javascript:alert(/XSS/) type=submit>'--></script>
October 15th 2019, 23:41:21.000	
October 15th 2019, 23:41:21.000	<SCRIPT>alert("XSS")</SCRIPT>>
October 15th 2019, 23:41:21.000	
October 15th 2019, 23:41:21.000	
October 15th 2019, 23:41:21.000	</script><script>alert('XSS');</script>
October 15th 2019, 23:41:21.000	
October 15th 2019, 23:41:21.000	
October 15th 2019, 23:41:21.000	
October 15th 2019, 23:41:21.000	','alert(String.fromCharCode(88,83,83))//','alert(String.fromCharCode(88,83,83))//";
October 15th 2019, 23:41:21.000	></SCRIPT>>'><SCRIPT>alert(String.fromCharCode(88,83,83))</SCRIPT>
October 15th 2019, 23:41:21.000	
October 15th 2019, 23:41:21.000	
October 15th 2019, 23:41:21.000	
October 15th 2019, 23:41:21.000	></SCRIPT>>'><SCRIPT>alert(String.fromCharCode(88,83,83))</SCRIPT>

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

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

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

10. THEHIVE ANALYST

10.1 INTRODUCTION:

Parameter	Description	Duration
Main Objective	This exercise introduces TheHive - platform supporting incident handling. Trainees are going to get familiar with TheHive, Cortex 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 to the exercise	20 minutes
	Task 1: Understanding general workflow of TheHive	15 minutes
	Task 2: Get familiar with TheHive interface	15 minutes
	Task 3: Performing an investigation of provided case by creating tasks, enriching data using Cortex analysers and discussion on obtained results.	40 minutes

In this part of the exercise, you will be introduced to TheHive⁵¹ – 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 MISP⁵².

TheHive uses Elasticsearch as its database. In the training environment, the Elasticsearch instance used by TheHive is storing its files on another Kubernetes⁵³ container. Such a setup allows restarting TheHive container without losing data (that normally happens to all changes that were made inside the container).

Cortex⁵⁴ 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.

10.2 TASKS:

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

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

and then

- `./start_exercise.sh.sh` (sudo pass: enisa)

⁵¹ <https://thehive-project.org>

⁵² <https://www.misp-project.org>

⁵³ <https://kubernetes.io>

⁵⁴ <https://github.com/TheHive-Project/CortexDocs>

And wait for the following message “Your environment is up and ready!”

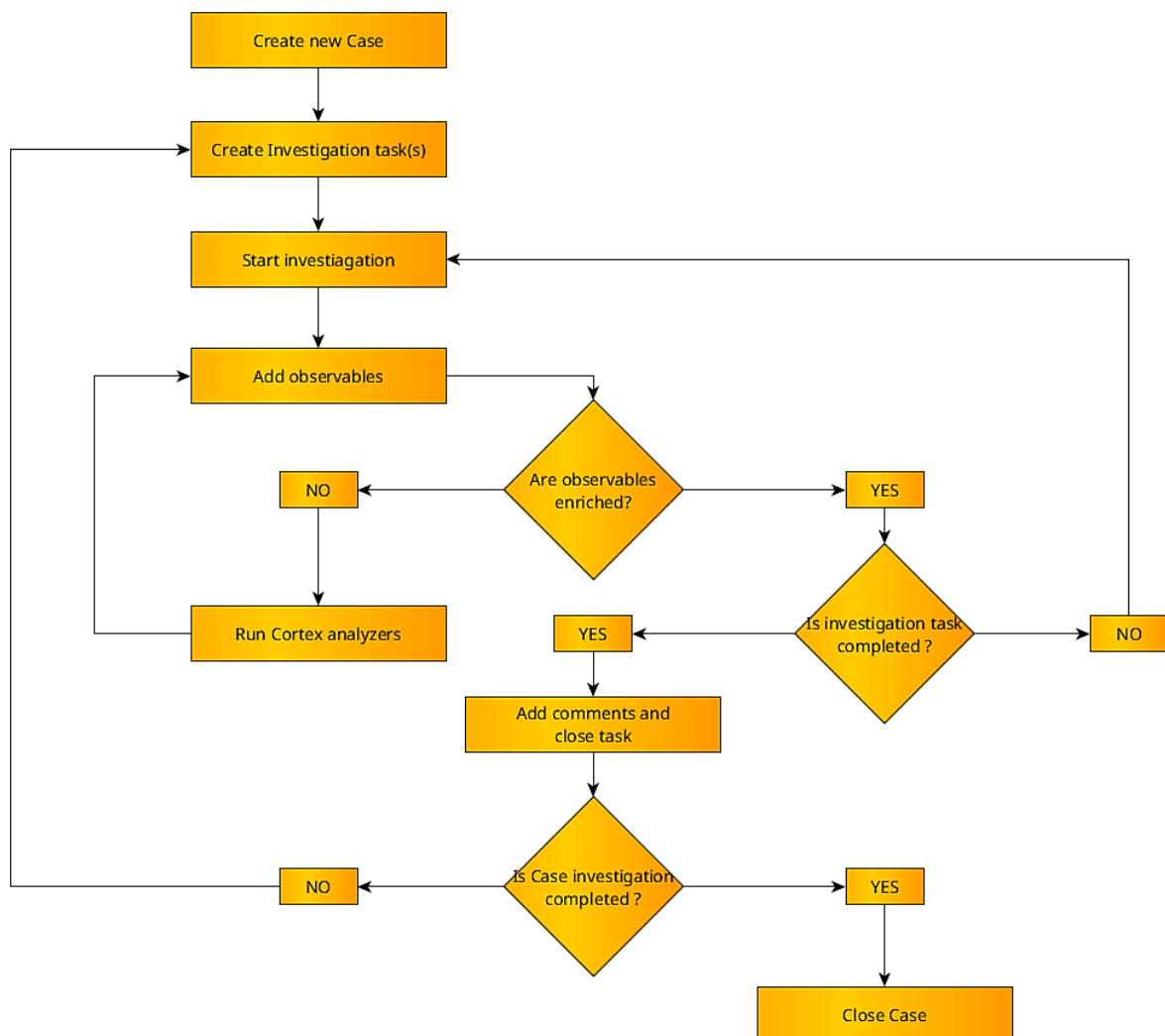
If you want to start the training all over again, you need to execute `./stop_exercise.sh` and then `./start_exercise.sh`

10.2.1 General workflow

Important concepts that will help you understand the workflow in TheHive include:

- **case** – it is the root object of investigation,
- **task** – one or more tasks can belong to a case.
- **observables** – added during the investigation, similar to MISP attributes, can be marked as Indicators of Compromise and sighted.
- **alerts** – security events, which can be imported e.g. from MISP

General workflow is shown on the following graph:



The idea shown here is that in order to perform investigation, you need to create tasks (those can be separated between analysts) and tasks should result in getting new information and this should be put into the TheHive (as observables or notes). Once new observables are added, we can enrich those using analysers to widen our understanding of the situation.

10.2.2 Let's get into UI!

Login to TheHive instance at thehive.enisa.europa.eu to an admin account (login admin, password admin).

Note: if you encounter SSL warning, you can ignore it, as this is a training environment. If you see a 50x error, wait a few seconds and refresh the page.

Let us focus on observables for a while. They can be of different types (IP, domain, hash, URL, ...) and some of them can be flagged as IoC (Indicator of Compromise) or additionally tagged.

Each observable must have defined TLP (Traffic light protocol⁵⁵) and a tag or description (or both). Observables can be exported in various formats, including MISP and analysed using Cortex Analysers.

10.2.3 Hands-on

Find an important alert in TheHive (triage). Alerts are coming from MISP events (because integration is configured and enabled).

The most relevant event corresponds to a report describing a campaign targeting the CSIRT's sector. Phishing email and other IoCs are included in the original event. Import it as a case by clicking the "Preview and import" icon on the right side. Then scroll down and click "Yes, Import" with the "Empty case" option.

Then create two tasks for investigating both IP addresses that are added as observables. To do so, go to the "Tasks" tab and create one by one by clicking on "Add Task".

One task will concern C&C IP address (Command and Control) and the second will be about IP addresses used for recon. You can fill the "Task group" field e.g. "data enrichment". Task groups can help better understanding what this particular task is about. Keep in mind, that those two tasks could be assigned to two analysts and performed simultaneously.

We will utilize Cortex to get more information about the incident observables in an easy way. Supporting datasets are provided: logs in Elasticsearch (containing information about who accessed our CSIRT website) and the ipasn database (matches particular IP addresses to the AS number that they are assigned to).

First, start a task concerning IP address used for recon (you can tell which one it is by reading a description when you put the cursor over IP). Click "Start" and run analysers against that IP. To do that, go to observables tab, click on [ip]: 215[.]148[.]86[.]190 and scroll down to the list of analysers. Then click on all red icons in "Actions" column to start analysis. Results can be obtained by clicking on the analysis date; json returned from the analyser script can be seen by clicking "Show raw report".

Discuss: What is the result of analysis? Did you obtain any additional information?

Add them as notes in the task, set up a "Has been sighted" and then close the task.

Next, start another task and proceed with the same steps as with the previous IP. Run analysers and see if there is any additional info. Pay close attention to retrieved the AS number. Add it as another observable as it is related to the case. To do that, go to observables tab, click "Add observables", select "other" from the type dropdown and paste "AS327712" at "Value" field.

⁵⁵ <https://www.us-cert.gov/tlp>

Write a short description and submit by clicking “Create observable(s)”. Now click on the ASN observable and check if there is a related case. If there is, check that case and try to gain more knowledge from it.

When you are done, add notes to the task and close it.

Conclude the investigation by exporting IoC’s to MISP by clicking “Share” and “Export” in a popup window.

Now you’re ready to close the case in TheHive by clicking the “Close” button in the Case header, selecting appropriate status, writing a brief summary and clicking “Close case”.



11. ARCHITECTURE AND TECHNICAL BACKGROUND

11.1 INTRODUCTION

Each exercise is contained in a single repository. The repository structure is presented below:

- **ADMIN** - scenarios for the admin part of the workshop
 - **Elasticsearch**
 - elasticsearch-bare - represents starting state of the Elasticsearch admin scenario
 - exercise/basics - resources for building the manual for the scenario
 - README.md - manual for the trainee
 - **IntelMQ**
 - intelmq-clean - represents starting state of the IntelMQ admin scenario
 - intelmq-populated - represents final state of the IntelMQ admin scenario
 - Scripts - helpers scripts for running honeypot services and data conversion
 - **MISP**
 - Data - data needed for exercises
 - misp-bare - represents starting state of the MISP admin scenario
 - misp-configured - represents final state of the MISP admin scenario
 - README.md - manual for the trainee
 - Reset-data.sh - script for resetting the state of the scenario
 - **TheHive**
 - Cortex-analyzers - directory to store Cortex analysers
 - Cortex-config - directory mapped to config directory of Cortex service, allows editing it from hosting system
 - Data - directory containing data for services used in TheHive scenario
 - thehive-bare - directory contains helm configuration of starting state of thehive admin scenario
 - thehive-config - directory mapped to config directory of TheHive service, allows editing it from hosting system
 - hive_bash.sh - allows to get shell inside TheHive pod
 - hive_logs.sh - allows to retrieve logs concerning TheHive application
 - reset_data.sh - brings exercise environment to initial state
 - restart_thehive.sh - restarts pod containing TheHive application
 - start_exercise.sh - script starting TheHive exercise environment
 - stop_exercise.sh - script stopping TheHive exercise environment
- **ANALYST** - scenarios for the Security Analyst part of the workshop
 - **IntelMQ**
 - Config - config files used by IntelMQ (preconfigured pipeline state)
 - exercise/basics - (pictures/screenshots used by the trainee manual)
 - intelmq - represents a starting state of the IntelMQ analyst scenario
 - Shared - shared directory for files used by the IntelMQ and analyst
 - README.md - manual for the trainee
 - **MISP**
 - Data - data needed for exercises
 - misp-bare - represents starting state of the MISP analyst scenario



- misp-configured - represents final state of the MISP analyst scenario
- README.md - manual for the trainee
- Reset-data.sh - script for resetting the state of the scenario
- **TheHive**
 - cortex-analyzers - directory to store cortex analysers
 - cortex-config - directory mapped to config directory of Cortex service, allows editing it from hosting system
 - data - directory containing data for services used in TheHive scenario
 - thehive-configured - directory contains helm configuration of starting state of TheHive analyst scenario
 - thehive-config - directory mapped to config directory of TheHive service, allows editing it from hosting system
 - hive_bash.sh - allows to get shell inside TheHive pod
 - hive_logs.sh - allows to retrieve logs concerning TheHive application
 - reset_data.sh - brings exercise environment to initial state
 - restart_thehive.sh - restarts pod containing TheHive application
 - start_exercise.sh - script starting TheHive exercise environment
 - stop_exercise.sh - script stopping TheHive exercise environment
- **Server-configuration** - configuration needed for setting up workshop framework on the machine
 - Ingress
 - README.md
- **README.md**

11.2 ARCHITECTURE

The whole infrastructure is based on Kubernetes⁵⁶ with help from Helm^{57,58} for managing packages, which we call scenarios.

For the Kubernetes cluster, we used microk8s,⁵⁹ which allows to easily setup a single-node Kubernetes cluster locally. Our goal here was to create a solution that can be adapted to work as a cloud solution without requiring a lot of changes.

Each of the scenarios is a single [helm chart](#). Chart represents one scenario which can be built on top of multiple systems. Systems that make up a scenario are supposed to be easily swappable between the scenarios.

A chart is organised as a collection of files inside of a directory. The directory name is the name of the chart (without versioning information).

11.3 ADDING A NEW SYSTEM

Adding a new system to the framework is a multiple step process.

- Find or create a Dockerfile for the system you want to setup and upload it to the Dockerhub
- Create a Kubernetes manifest file describing the system. You will probably need to specify at least Deployment and Service, maybe an Ingress as well. (Check existing files for more information).
- Find out which paths inside the Docker container are used for database/config and can be used as a way to create multiple states of the system. e.g. in MISP we used two paths for customization - `"/var/lib/mysql"` and `"/var/www/MISP/app/Config"`.
- Create volume mounts in the manifest file from paths discovered in the previous step.

⁵⁶ <https://kubernetes.io/>

⁵⁷ <https://helm.sh/>

⁵⁸ <https://github.com/helm/helm>

⁵⁹ <https://microk8s.io/>

- Copy data from the Docker container outside to successfully mount them as hostPaths.
- Create templates in places of the host paths to allow for changing them with values.yaml from Helm.
- (Optionally) If you would like to add new data on top of that contained in the database files, you can add a post-install hook to the helm chart. You can read more about them [here](#).

11.4 ADDING A NEW SCENARIO

To add new scenario we have to create a new chart. This can be done by typing the following command:

- `helm create chartname`

This should create a new directory with a helm chart directory structure.

- Chartname
 - Values.yaml
 - Chart.yaml
 - templates/
 - charts/

[This resource](#) explains the purpose of each of the files contained in the directory.

The most important part of this structure is the templates directory. It is the place where all Kubernetes manifests (.yaml files) are stored.

The next step is to fill the templates folder with yamls representing the system you would like to setup.

If you want to add a new system look at the “Adding new system” chapter first. This is the place where you would add your Kubernetes manifest with abstracted out variables (all of the jinja syntax you can see below).

As an example, we present MISP Kubernetes manifest file used by our scenarios:

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: {{ template "misp.fullname" . }}
  labels:
    app.kubernetes.io/instance: {{ .Release.Name }}
    helm.sh/chart: {{ .Chart.Name }}-{{ .Chart.Version }}
    app.kubernetes.io/name: {{ template "misp.name" . }}
spec:
  replicas: {{ .Values.replicaCount }}
  template:
    metadata:
      labels:
        app.kubernetes.io/name: {{ template "misp.name" . }}
```

```
app.kubernetes.io/instance: {{ .Release.Name }}
spec:
  containers:
    - name: {{ template "misp.name" . }}
      image: "{{ .Values.image.repository }}:{{ .Values.image.tag }}"
      imagePullPolicy: {{ .Values.image.pullPolicy }}
      ports:
        - name: https
          containerPort: 443
          protocol: TCP
      volumeMounts:
        - mountPath: /var/lib/mysql
          name: misp-mysql-volume
        - mountPath: /var/www/MISP/app/Config
          name: misp-config-volume
  volumes:
    - name: misp-mysql-volume
      hostPath:
        path: {{ .Values.volumes.mysqlPath | quote }}
        type: Directory
    - name: misp-config-volume
      hostPath:
        path: {{ .Values.volumes.configPath | quote }}
        type: Directory
```

To allow multiple states to be used for each system, we opted for mounting configuration/database files from configurable locations.

Locations for database and config files are abstracted out to the values.yaml file.

NOTE: *To create a new system state, copy existing one and create new chart.*

Now run this chart by typing `helm install newchart`.

You can play with the running system and change its state now, all the changes are going to be saved on disk. This is how you create new state from previous one.

After putting together your chart, you can test it out with Helm install chartname

11.5 DEBUGGING ESSENTIALS

This section should help in case of difficulties with training environment.

For setting up instances of tools, we are using helm so you need to get familiar with basic commands to work with it.

To list currently installed charts, in terminal type: “helm ls”

In addition, in case of running TheHive training you should get something like:

```
$ helm ls
```

NAME	REVISION	UPDATED	STATUS	CHART	APP VERSION	NAMESPACE
peeking-emu	1	Tue Dec 3 19:15:59 2019	DEPLOYED	thehive-configured-0.1.0	1.0	default

In order to get more details about a particular helm chart use Helm status <name of the deployment>, eg:

```
$ helm status idolized-snake
LAST DEPLOYED: Tue Dec 3 13:34:10 2019
NAMESPACE: default
STATUS: DEPLOYED

RESOURCES:
==> v1/ConfigMap
NAME      AGE
es-config 115m

==> v1/Deployment
NAME      AGE
hive-elastic 115m
thehive   115m
thehive-cortex 115m

==> v1/Pod(related)
NAME      AGE
hive-elastic-84c7d478f4-6rwnj 115m
idolized-snake-misp2-5774fdcb86-vkjjk 115m
thehive-5cb485c7bb-cc7f5 115m
thehive-cortex-58777c676-lkfs4 115m

==> v1/Service
NAME      AGE
cortex-service 115m
elastic-service 115m
misp2-service 115m
thehive-service 115m

==> v1beta1/Deployment
NAME      AGE
idolized-snake-misp2 115m

==> v1beta1/Ingress
NAME      AGE
idolized-snake-misp2 115m
ingress-cortex 115m
ingress-thehive 115m
```

As any helm chart is a set of running containers, above you can see a list of running pods and information about them.

To run any of the exercises, the user needs to type a “helm install <chart name> command.

If something goes wrong (eg. lost internet connectivity during image download) you may need to delete the chart by helm delete <chart name>

You can get a list of containers also by running “kubectl get pods”.

Another useful command is “kubectl version”. It not only shows versions of both client and server Kubernetes version but to check server version it also checks connectivity between the two. If in

server version, you get an error about TLS handshake timeout or Connection refused, you may need to restart microk8s service as described below.

If you want to get details on what is going on inside of the particular container, you can check it with “kubectl describe pod <name of the pod>”

In case you need to see the logs of an application running in a pod, execute “kubectl log <name of the pod>”.

To get bash command line inside of the container (e.g for debugging purposes) execute

```
“kubectl exec -it <name of the pod> bash”
```

11.6 POSSIBLE ERROR MESSAGES:

When you try to start any of the training environments, you can get error like below:

11.6.1 Error type 1

could not find a ready tiller pod

That error means that helm module responsible for chart management and usually takes some time to start up after booting the system. After few minutes helm should work properly.

11.6.2 Error type 2

release mewing-bobcat failed: configmaps "es-config" already exists

Probably some helm chart are up. Check that by “helm ls” and “helm delete <release name>”

11.6.3 Error type 3

get http://localhost:8080/api/v1/namespaces/kube-system/pods?labelSelector=app3Dhelm%2Cname%3Dtiller: dial tcp 127.0.0.1:8080: connect: connection refused

Why? The microk8s service could not be loaded properly. To diagnose that, run “microk8s.inspect” and see the results. You will see “self-diagnosis test” and probably one of the services be in FAIL state, eg.”FAIL: Service snap.microk8s.daemon-apiserver is not running”.

To get this fixed run microk8s.stop and then microk8s.start. If that does not help, update system

(<https://github.com/ubuntu/microk8s/issues/496>)

11.6.4 Error type 4

incompatible versions client[vx.x.x] server[v x.x.x] (x-ed version numbers)

Our Kubernetes templates use v1-beta API of Kubernetes and both server and client cannot be more recent than 1.15/stable (as there is no backward compatibility in 1.16). VM is prepared with this version but in case of accidental upgrade, you need to un-install microk8s and kubectl and then install them according to server-setup guide.

12. BIBLIOGRAPHY - REFERENCES

ENISA, Actionable Information for Security Incident Response, 2014.

https://www.enisa.europa.eu/topics/csirt-cert-services/reactive-services/copy_of_actionable-information

MISP - Threat Sharing. <https://www.circl.lu/doc/misp/>

Incident Handling Automation. <https://www.enisa.europa.eu/topics/csirt-cert-services/community-projects/incident-handling-automation>

ENISA, Proactive detection of security incidents II - Honeypots, 2012.

<https://www.enisa.europa.eu/publications/proactive-detection-of-security-incidents-ii-honeypots>

Information sharing and cooperation enabled by GDPR, Version: 1.1, January 2018.

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

IntelMQ: Data Harmonization. <https://github.com/certtools/intelmq/blob/master/docs/Data-Harmonization.md>

REGULATION (EU) 2016/679 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). <https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679&from=EN> .



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