

# MISP API and Automation Workshop

Tutorial and Hands-On

Alexandre Dulaunoy & Christian Studer

MISP Project

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



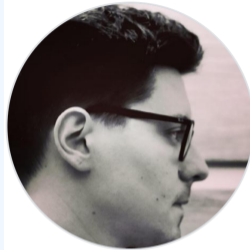


<https://link.infini.fr/metz-training>



**Alexandre Dulaunoy**

adulau



**Christian Studer**

chris3d

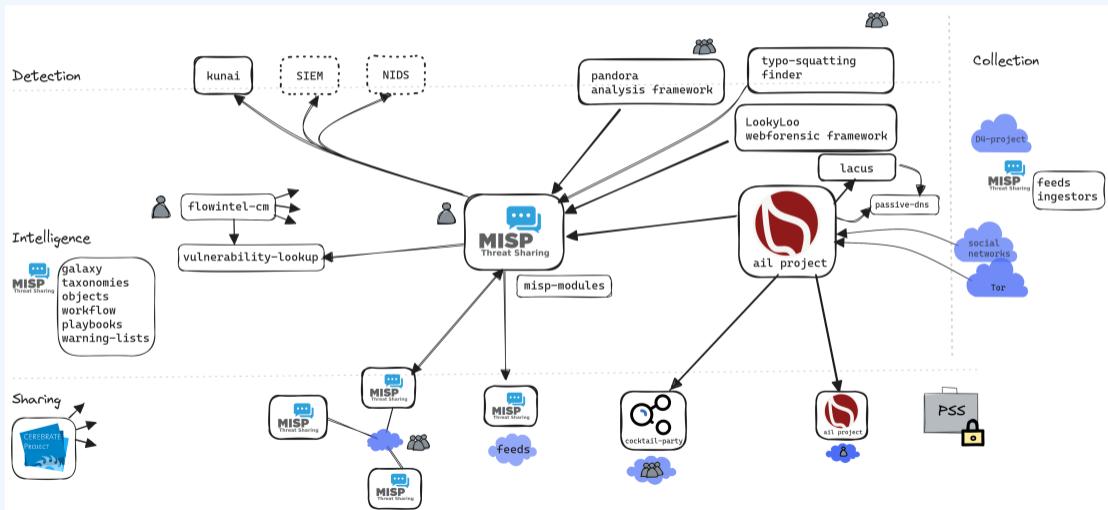


1. MISP API / PyMISP
  - ▶ Demo with examples
  - ▶ 3 Hands-on exercises
2. PubSub channels (ZeroMQ)
  - ▶ Demo
3. misp-modules
4. MISP Workflows
  - ▶ Fundamentals
  - ▶ Demo with examples
  - ▶ Usage and Plugins

This workshop requires some basic MISP knowledge



# INTERCONNECTING EVERYTHING



# MISP API / PyMISP

**Objective:** Get to know how to use the MISP API PyMISP



1. Generate an API key
2. Rest Client overview
3. OpenAPI specifications <sup>1</sup>
4. MISP API Overview notebook <sup>2</sup>
5. PyMISP Overview notebook <sup>3</sup>

---

<sup>1</sup><https://www.misp-project.org/openapi/>

<sup>2</sup><https://github.com/MISP/misp-training/blob/main/a.7-rest-API/Training%20-%20Using%20the%20API%20in%20MISP.ipynb>

<sup>3</sup><https://github.com/MISP/PyMISP/blob/main/docs/tutorial/FullOverview.ipynb>



# MISP API / PYMISP - HANDS-ON EXERCISE 1

**Objective:** Practise on creating Data via the API



Create an Event with the following:

1. 3 Attributes

- ▶ **ip-dst:** 1.2.3.4
- ▶ **domain:** evil.com
- ▶ **filename:** evil.exe

2. 1 Object

- ▶ Object type: **domain-ip**
- ▶ Attribute 1: **ip:** 4.3.2.1
- ▶ Attribute 2: **domain:** foobar.baz
- ▶ Attribute 3: **text:** Classified information

3. Change distribution of Object's Attribute 3 **text** to your org-only

4. Tag Attribute 1 **ip-dst** with tlp:green

Update the Event with the following:

1. Tag the Event with `tlp:clear`
2. Attach the cluster Energy from the Sector Galaxy to the Event
3. Tag Object's Attribute 3 **text** with `tlp:amber`
4. Publish the Event

# MISP API / PyMISP - HANDS-ON EXERCISE 2

**Objective:** Practise on filtering data



Create search queries to:

1. Get all Attributes that were published in the past 48 hours
2. Get all Attributes
  - ▶ of type ip-src and ip-dst
  - ▶ that were changed in the past 48
  - ▶ meant for protective tools
  - ▶ in the CSV format
3. Get the first page (20 / page) of Attributes
  - ▶ that **have a tlp** marking
  - ▶ but not tlp:amber, tlp:amber+strict or tlp:red
4. How many Events do we have
  - ▶ labelled Attack Pattern :: Phishing - T1566 ?

- MISP standard export modules (exposed via REST's `returnFormat`)
  - ▶ Many internal ones such as JSON, text, CSV, YARA, netfilter, etc.
  - ▶ `misp-stix`<sup>4</sup>
- MISP-module Export modules
- → primer on creating new exports

---

<sup>4</sup><https://github.com/MISP/misp-stix>

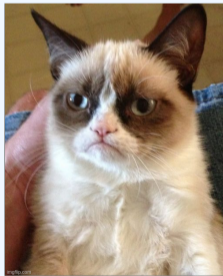
**Objective:** Introduction to `misp-playbooks`<sup>5</sup>

---

<sup>5</sup><https://github.com/MISP/misp-playbooks>

# PUBSUB CHANNELS (ZEROMQ) - FUNDAMENTALS

**Objective:** Learn how to setup realtime automation using the ZeroMQ channel





## 1. What is ZeroMQ?





- ▶ *N-to-N Asynchronous message-processing tasks*
- ▶ *Publisher (MISP) and consumer (scripts)*
- Demo: `tools/misp-zmq/sub.py`

## 2. Configuring ZeroMQ in MISP

## 3. Integrating with the ZeroMQ of MISP

# ZEROMQ CHANNEL - CONFIGURATION

## Server Settings & Maintenance

Overview MISP (24 ) Encryption (5) Proxy (5) Security (8 ) Plugin (73 ) SimpleBackgroundJobs Correlations **new** Diagnostics Manage files 

ZeroMQ

Filter the table(s) below

Optional	Plugin.ZeroMQ_enable	true	Enables or disables the pub/sub feature of MISP. Make sure that you install the requirements for the plugin to work. Refer to the installation instructions for more information.	
Optional	Plugin.ZeroMQ_host	0.0.0.0	The host that the pub/sub feature will use.	
Optional	Plugin.ZeroMQ_port	50000	The port that the pub/sub feature will use.	Value not set.
Optional	Plugin.ZeroMQ_event_notifications_enable	true	Enables or disables the publishing of any event creations/edits/deletions.	
Optional	Plugin.ZeroMQ_object_notifications_enable	true	Enables or disables the publishing of any object creations/edits/deletions.	
Optional	Plugin.ZeroMQ_object_reference_notifications_enable	true	Enables or disables the publishing of any object reference creations/deletions.	
Optional	Plugin.ZeroMQ_attribute_notifications_enable	true	Enables or disables the publishing of any attribute creations/edits/soft deletions.	
Optional	Plugin.ZeroMQ_tag_notifications_enable	true	Enables or disables the publishing of any tag creations/edits/deletions as well as tags being attached to / detached from various MISP elements.	
Optional	Plugin.ZeroMQ_sighting_notifications_enable	true	Enables or disables the publishing of new sightings to the ZMQ pubsub feed.	
Optional	Plugin.ZeroMQ_user_notifications_enable	true	Enables or disables the publishing of new/modified users to the ZMQ pubsub feed.	
Optional	Plugin.ZeroMQ_organisation_notifications_enable	true	Enables or disables the publishing of new/modified organisations to the ZMQ pubsub feed.	
Optional	Plugin.ZeroMQ_audit_notifications_enable	true	Enables or disables the publishing of log entries to the ZMQ pubsub feed. Keep in mind, this can get pretty verbose depending on your logging settings.	
Optional	Plugin.ZeroMQ_warninglist_notifications_enable	false	Enables or disables the publishing of new/modified warninglist to the ZMQ pubsub feed.	

# ZEROMQ CHANNEL - INTEGRATION

---

```
1 # Imports libraries
2
3 socks = dict(poller.poll(timeout=None))
4 while True:
5     if socket in socks and socks[socket] == zmq.POLLIN:
6         message = socket.recv()
7         topic, s, m = message.decode('utf-8').partition(" ")
8         handleMessage(topic, m)
9         time.sleep(1)
10
11 def handleMessage(topic, message):
12     if topic == "misp_json_event":
13         handleEvent(message)
14     if topic == "misp_json_attribute":
15         handleAttribute(message)
16     ...
```

---

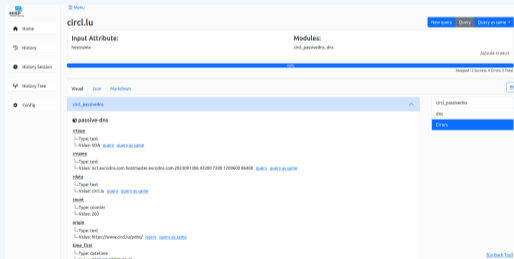
---

```
1 def handleEvent(message)(event):
2     cnt_attr = len(event['Attribute'])
3     cnt_obj = len(event['Object'])
4     url = misp_url + '/events/view/' + event['id']
5     message_short = f"""
6         New MISP event '{event['info']}'
7         with {cnt_attr} attributes, {cnt_obj} objects.
8     """
9
10    # Send the message
11    client = slack.WebClient(token=SLACK_TOKEN)
12    client.chat_postMessage(
13        text=message_short,
14    )
```

---

# MISP MODULES

- MISP modules, with more than 200 available, are tools and functionalities designed to enhance and extend the capabilities of the MISP platform.
- These modules are now standalone.<sup>6</sup>
- The standard format used by MISP modules is consistent with the MISP core format.



<sup>6</sup><https://www.misp-project.org/2024/03/12/Introducing-standalone-MISP-modules.html>

# MISP WORKFLOWS - PRIMER

**Objective:** Learn how to use MISP Workflows



## MISP API / PyMISP

- Needs **CRON Jobs** in place
- Potentially heavy for the server
- **Not realtime**



## PubSub channels

- After the actions happen: **No feedback** to MISP (it's only a publish channel)
  - **Tougher to put in place** & to **share**
  - Full integration amounts to **develop a new tool**
- No way to **prevent** behavior
- Difficult to setup **hooks** to execute callbacks



- **Prevent** default MISP behaviors to happen
  - ▶ Prevent **publication of events** not passing sanity checks
  - ▶ Prevent **querying** thrid-party **services** with sensitive information
  - ▶ ...
- **Hook** specific actions to run callbacks
  - ▶ **Automatically run** enrichment services
  - ▶ Modify data on-the-fly: False positives, enable CTI-Pipeline
  - ▶ Send notifications in a chat rooms

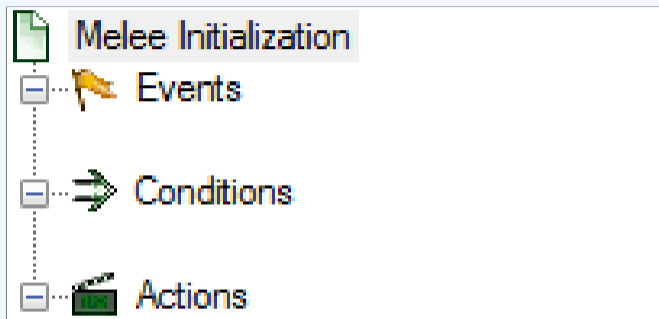


- **Notification** on specific actions
  - ▶ New events matching criteria
  - ▶ New users
  - ▶ Automated alerts for high-priority IOCs
- **Extend** existing MISP behavior
  - ▶ Push data to another system
  - ▶ Automatic enrichment
  - ▶ Sanity check to block publishing / sharing
- **Hook** capabilities
  - ▶ Assign tasks and notify incident response team members
  - ▶ Run curation pipeline
- ...

# WORKFLOW - FUNDAMENTALS

**Objective:** Start with the foundation to understand the basics






1. An **event** happens in MISP
2. Check if all **conditions** are satisfied
3. Execute all **actions**
  - ▶ May prevent MISP to complete its original event

## Events

- New MISP Event
- Attribute has been saved
- New discussion post
- New user created
- Query against third-party services
- ...

 Supported events in MISP are called **Triggers**

 A **Trigger** is associated with **1-and-only-1 Workflow**

# TRIGGERS CURRENTLY AVAILABLE

Currently 14 triggers can be hooked. 5 being Blocking.

## Triggers

List the available triggers that can be listened to by workflows.






















































Missing a trigger? Feel free to open a [Github issue!](#)

[Documentation and concepts](#)

« previous

next »

All attribute event object others post user Blocking Enabled Disabled

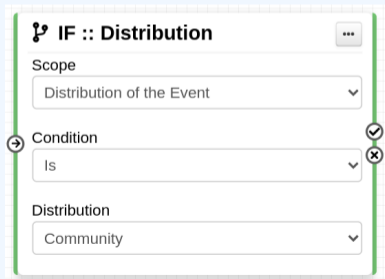
Trigger name	Scope	Trigger overhead	Run counter	Blocking Workflow	MISP Core format	Workflow ID	Last Update	Debug enabled	Enabled	Actions
 Attribute After Save	attribute	high 	83	×	✓	160	2022-08-03 09:00:41	<input type="checkbox"/>	×	   
 Enrichment Before Query	others	low	1154	✓	✓	162	2022-10-17 12:35:57	<input type="checkbox"/>	✓	   
 Event After Save	event	high 	49	×	✓	175	2022-10-14 13:32:01	<input type="checkbox"/>	✓	   
 Event After Save New	event	low	5	×	✓	182	2022-10-17 09:12:14	<input checked="" type="checkbox"/>	✓	   
 Event After Save New From Pull	event	low	6	×	✓	183	2022-10-17 09:01:36	<input checked="" type="checkbox"/>	✓	   
 Event Publish	event	low	126	✓	✓	180	2022-10-13 10:42:53	<input checked="" type="checkbox"/>	✓	   
 Object After Save	object	high 	35	×	✓	161	2022-08-05 07:12:52	<input type="checkbox"/>	×	   
 Post After Save	post	low	36	×	×	176	2022-07-28 13:59:51	<input type="checkbox"/>	×	   
 User After Save	user	low	0	×	×	181	2022-08-05 07:19:46	<input type="checkbox"/>	×	   
 User Before Save	user	low	42	✓	×	158	2022-07-28 14:00:32	<input type="checkbox"/>	×	   

# WHAT KIND OF CONDITIONS?

## ⇒ Conditions

- A MISP Event is tagged with `tlp:red`
- The distribution of an Attribute is a sharing group
- The creator organisation is `circ1.lu`
- Or any other **generic** conditions

❓ These are also called **Logic modules**























The screenshot shows a configuration window for a logic module titled "IF :: Distribution". It contains three dropdown menus:

- Scope:** Set to "Distribution of the Event".
- Condition:** Set to "Is".
- Distribution:** Set to "Community".

There are control icons on the right side: a checkmark next to the Condition dropdown and an 'X' next to the Distribution dropdown. A plus sign icon is visible on the left side of the Condition dropdown.

# WORKFLOW - LOGIC MODULES

- ➡ 10 **logic** modules: Allow to redirect the execution flow.
  - ▶ IF conditions
  - ▶ Delay execution

All		Action	Logic	misp-module	Custom	Blocking	Enabled	Disabled	Enter value to search	Filter	×
<input type="checkbox"/>	Module name	Type	Blocking	MISP Core format	misp-module	Custom	Enabled	Actions			
<input type="checkbox"/>	 Blueprint logic module	logic	×	×	×	✓	×	▶ 			
<input type="checkbox"/>	 Concurrent Task	logic	×	×	×	×	✓	■ 			
<input type="checkbox"/>	 IF :: Distribution	logic	×	✓	×	×	✓	■ 			
<input type="checkbox"/>	 Filter :: Generic	logic	×	×	×	×	×	▶ 			
<input type="checkbox"/>	 Filter :: Remove filter	logic	×	×	×	×	×	▶ 			
<input type="checkbox"/>	 IF :: Generic	logic	×	×	×	×	✓	■ 			
<input type="checkbox"/>	 IF :: Organisation	logic	×	✓	×	×	✓	■ 			
<input type="checkbox"/>	 IF :: Published	logic	×	✓	×	×	✓	■ 			
<input type="checkbox"/>	 IF :: Tag	logic	×	✓	×	×	✓	■ 			
<input type="checkbox"/>	 IF :: Threat Level	logic	×	×	×	×	×	▶ 			

# WHAT KIND OF ACTIONS?



## Actions


- Send an email notification
- Perform enrichments
- Send a chat message on MS Teams
- Attach a local tag
- ...

? These are also called **Action modules**

The screenshot shows a configuration window for a 'Send Mail' action. The title is 'Send Mail' with an envelope icon and a three-dot menu icon. Below the title is the instruction 'Allow to send a Mail to a list or recipients'. The 'Recipients' section has a dropdown menu currently showing 'All accounts'. The 'Mail template subject' section has a text input field containing 'I'm the mail subject!'. The 'Mail template body' section has a text input field containing 'And I'm the body!'. There are circular arrows on the left and right sides of the subject and body sections, indicating they can be moved or reordered.



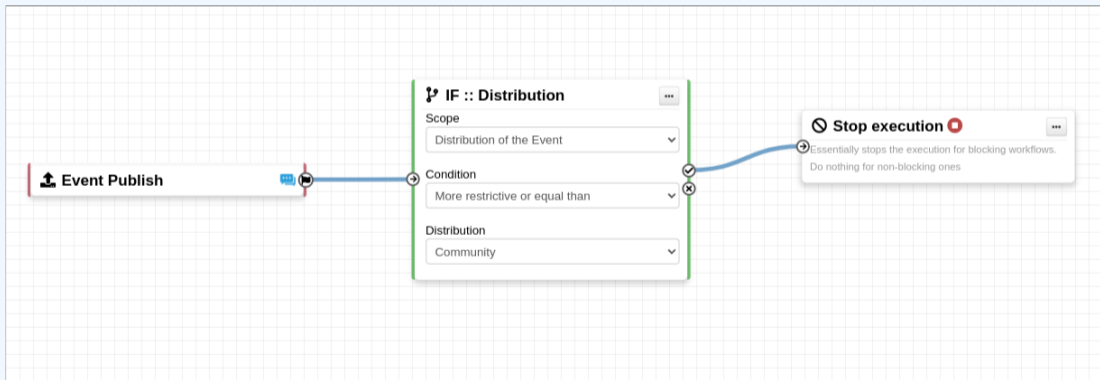
# WORKFLOW - ACTION MODULES

-  23 **action** modules: Allow to executes operations
  - ▶ Tag operations
  - ▶ Send notifications
  - ▶ Webhooks & Custom scripts

All <b>Action</b> Logic misp-module Custom Blocking Enabled Disabled								Enter value to search	Filter X
<input type="checkbox"/>	Module name	Type	Blocking	MISP Core format	misp-module	Custom	Enabled	Actions	
<input type="checkbox"/>	* Attach enrichment	action	x	✓	x	x	✓	■ 👁	
<input type="checkbox"/>	🔗 Attribute edition operation	action	x	✓	x	x	✓	■ 👁	
<input type="checkbox"/>	🔗 Attribute IDS Flag operation	action	x	✓	x	x	✓	■ 👁	
<input type="checkbox"/>	🏗️ Blueprint action module	action	x	x	x	✓	✓	■ 👁	
<input type="checkbox"/>	* Enrich Event	action	x	✓	x	x	✓	■ 👁	
<input type="checkbox"/>	📧 mattermost	action	x	x	✓	x	✓	■ 👁	
<input type="checkbox"/>	🗣️ MS Teams Webhook	action	x	x	x	x	✓	■ 👁	
<input type="checkbox"/>	🔗 Push to ZMQ	action	x	x	x	x	✓	■ 👁	
<input type="checkbox"/>	✉️ Send Log Mail	action	x	x	x	x	x	▶ 👁	
<input type="checkbox"/>	✉️ Send Mail	action	x	x	x	x	✓	■ 👁	
<input type="checkbox"/>	> Solunk HEC export	action	x	✓	x	x	x	▶ 👁	
			x		x	x	✓	■ 👁	

# WHAT IS A MISP WORKFLOW?

- Sequence of all nodes to be executed in a specific order
- Workflows can be enabled / disabled
- A Workflow is associated to **1-and-only-1 trigger**



# WORKFLOW EXECUTION FOR EVENT PUBLISH



An Event is about to be published

- ▶ The workflow for the event-publish trigger starts



Conditions are evaluated

- ▶ They might change the path taken during the execution



Actions are executed

- ▶ **success**: Continue the publishing action

```
execute_workflow Finished executing workflow for trigger `event-publish` (180). Outcome: success
```

- ▶ **failure** | **blocked**: Stop publishing and log the reason

```
execute_workflow Execution stopped.  
Node `stop-execution` (8) from Workflow `Workflow for trigger event-publish` (180) returned the following error: Execution stopped
```

# BLOCKING AND NON-BLOCKING

Two types of workflows:

## ❑ Blocking Workflows

- ▶ Can prevent / block the original event to happen
- ▶ If a **blocking module** ❑ blocks the action

## ✔ Non blocking Workflows execution outcome has no impact

- ▶ No way to prevent something that happened in the past



Currently 47 built-in modules.

- **Trigger** module (11): built-in **only**
  - ▶ Get in touch if you want more
- **Logic** module (10): built-in & **custom**
- **Action** module (15): built-in & **custom**

# SOURCES OF WORKFLOW MODULES (1)

- Built-in **default** modules

- ▶ Part of the MISP codebase
- ▶ Get in touch if you want us to increase the selection (or do a PR!)



# SOURCES OF WORKFLOW MODULES (2)


## User-defined **custom** modules

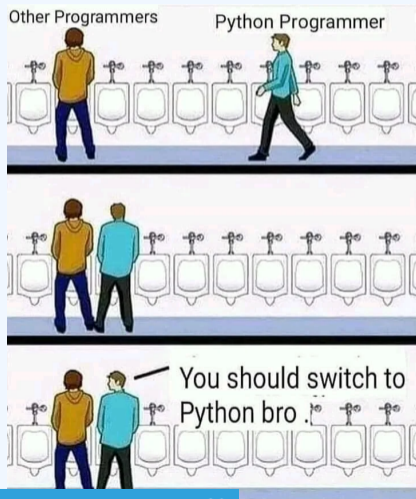
- Written in PHP 🖱️
- Extend existing modules
- MISP code reuse



# SOURCES OF WORKFLOW MODULES (3)

Modules from the **misp-module**  **enrichment service**

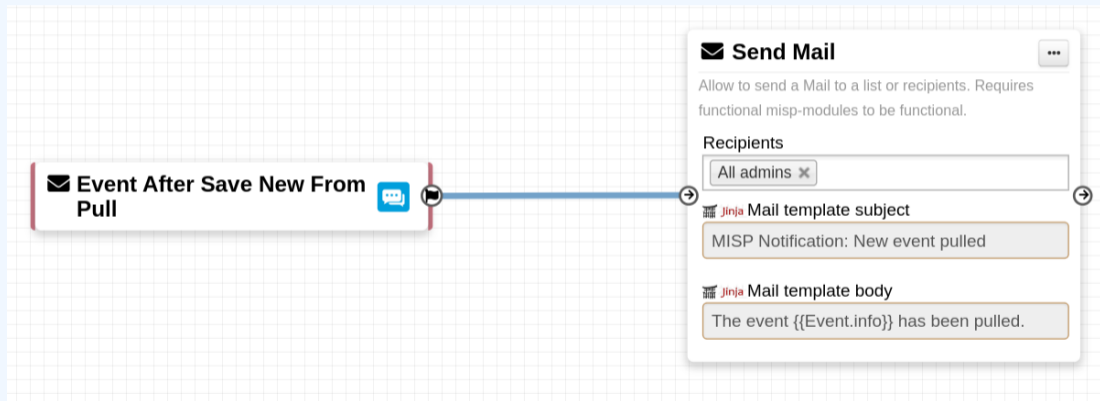
- Written in Python 
- Can use any python libraries
- Plug & Play





# DEMO BY EXAMPLES 1

Send an email to **all** when a new event has been pulled



## DEMO BY EXAMPLES 2

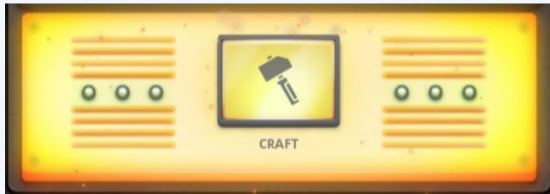
Block queries on 3rd party services when **tlp:red** or **PAP:red**

- **tlp:red**: For the eyes and ears of individual recipients only
- **PAP:RED**: Only passive actions that are not detectable from the outside



# WORKFLOW - GETTING STARTED

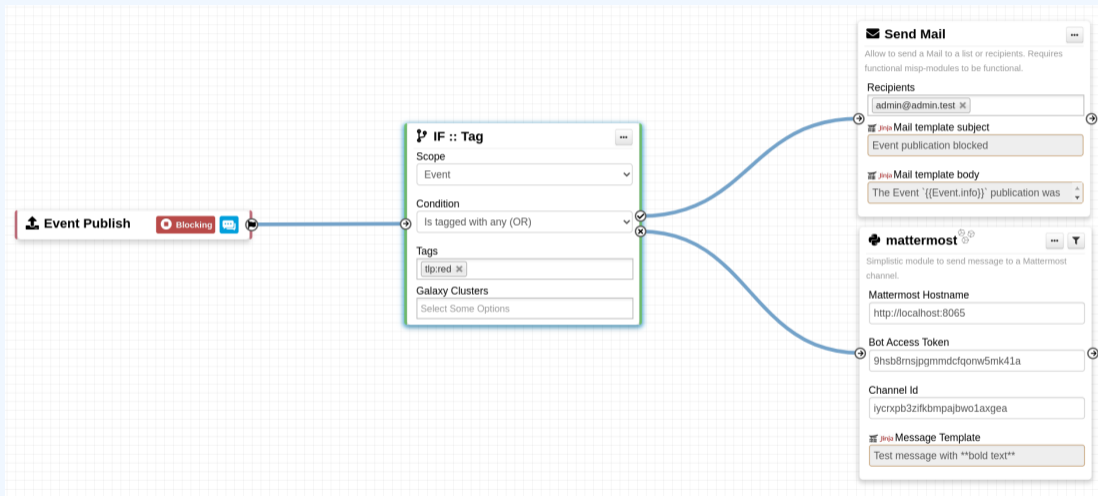
**Objective:** Learn how to build a Workflow



Let's build the following Workflow:

1. Prevent event publication if **tlp:red** tag is attached
2. Send a mail to `admin@admin.test` about potential data leak
3. Otherwise, send a notification on **Mattermost, MS Teams, Telegram, ...**

# CREATING A WORKFLOW WITH THE EDITOR



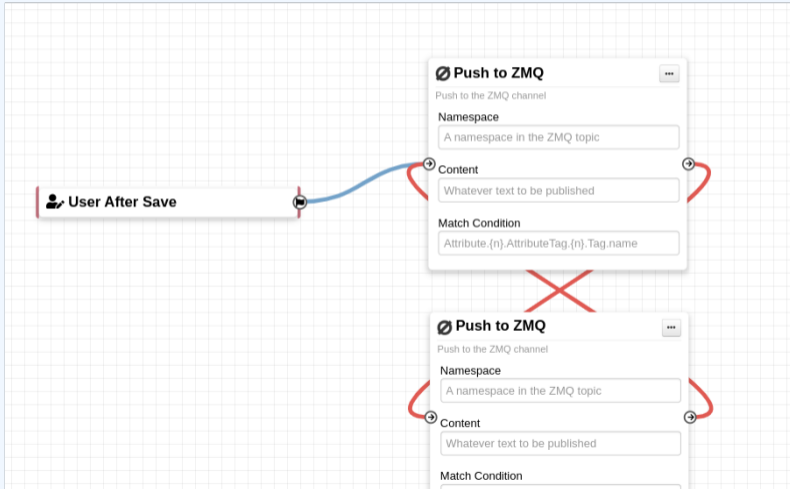
# CONSIDERATIONS WHEN WORKING WITH WORK-FLOWS

**Objective:** Overview of some common pitfalls



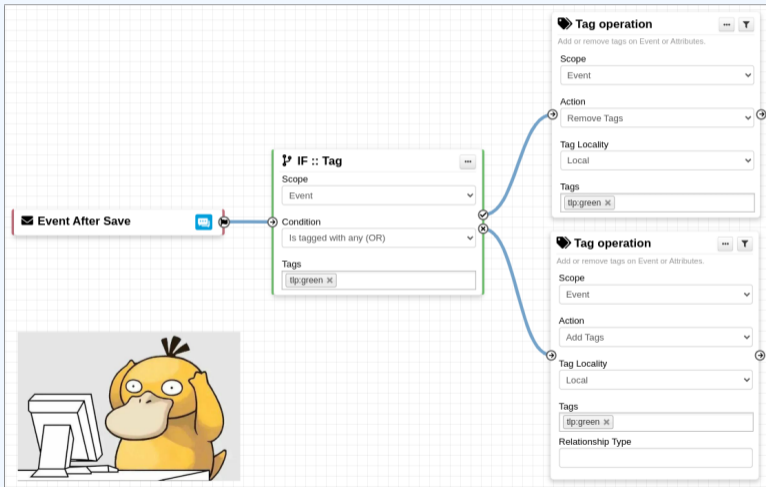
# WORKING WITH THE EDITOR - OPERATIONS NOT ALLOWED

Execution loop are not authorized



# RECURSIVE WORKFLOWS

! Recursion: If an action re-run the workflow










# EXERCICES

Try to build it in the training instance. **⚠ Do not save it! ⚠**

1. Allow **Publishing** Events only tagged with `tlp:clear`, `tlp:white` or `tlp:green` tags
2. **Replace** the tag `tlp:white` by `tlp:clear`

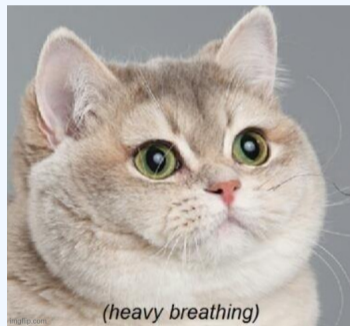
- Data filtering & Path filtering
-  Extended MISP Core format
-  Blueprints
-  Live debugging & stateless execution
-  Extending with plugins
-  Jinja templating
- Concurrent Tasks
- Annotated frames

# SHOULD I MIGRATE TO MISP WORKFLOWS

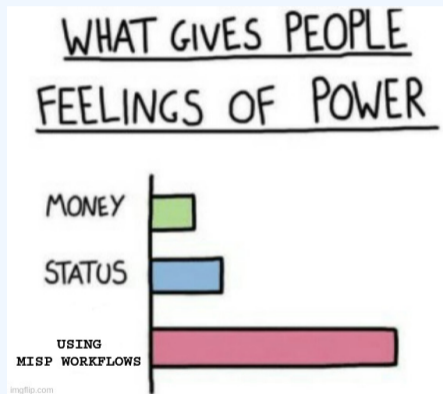
I have automation in place using the **API / ZMQ**. Should I move to Workflows?

- I (have/am planning to create) a curation pipeline using the API, should I port them to workflows?
  - ▶ **No** in general, but WF can be used to start the curation process
- What if I want to **block** some actions
  - ▶ Put the blocking logic in the WF, the remaining outside
- Currently, workflows with **lots of node are not encouraged**
  - ▶ > 20 nodes, mainly for readability
- Bottom line is **Keep it simple**

- More 📁 modules
- More ➡ modules
- More 🖱️ triggers
- More documentation
- Recursion prevention system
- On-the-fly data override?



- Designed to **quickly** and **cheaply** integrate MISP in CTI pipelines
- **Beta** Feature unlikely to change. But still..
- Waiting for feedback!
  - ▶ New triggers?
  - ▶ New modules?
  - ▶ What's achievable



- In MISP, you have **more than one way to do it**, and that's applicable to the API/automation part.
- Pick the one that fits your use-case and constraints.
- Don't be afraid to check with us if you have specific requirements.
- Don't forget the performance implications of automation (be nice with other MISP instances).