

AUTOMATION WITH WORKFLOWS IN MISP

SHORT VERSION

SAMI MOKADDEM

MISP PROJECT

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



1. Automation in MISP
2. MISP Workflows
 - ▶ Fundamentals
 - ▶ Demo with examples
 - ▶ Using the system
 - ▶ How it can be extended



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
- 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



- **Visual** dataflow programming
- **Drag & Drop** editor
- Flexible **Plug & Play** system
- **Share** workflows, **debug** and **replay**

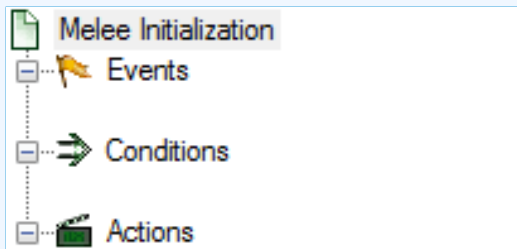
- **Notification** on specific actions
 - ▶ New events matching criteria
 - ▶ New users
 - ▶ Automated alerts for high-priority IOCs
- **Extend** existing MISP behavior
 - ▶ Push data to another system
 - ▶ Automatic enrichment
 - ▶ Sanity check to block publishing / sharing
 - ▶ Curation pipelines
-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

WORKFLOW - FUNDAMENTALS

Objective: Start with the foundation to understand the basics



HOW DOES IT WORK




1. An **event** happens in MISP
2. *(optional)* 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 11 triggers can be hooked. 3 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 log 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 ?	110	×	✓	160	2023-09-14 06:54:37	<input type="checkbox"/>	×	▶️ ⚙️ 📄 🔗
✳️ Enrichment Before Query	others	low	2226	✓	✓	162	2023-10-09 07:56:42	<input type="checkbox"/>	✓	▶️ ⚙️ 📄 🔗
📧 Event After Save	event	high ?	191	×	✓	175	2023-10-02 14:55:19	<input type="checkbox"/>	×	▶️ ⚙️ 📄 🔗
📧 Event After Save New	event	low	7	×	✓	182	2023-03-16 14:05:07	<input type="checkbox"/>	×	▶️ ⚙️ 📄 🔗
📧 Event After Save New From Pull	event	low	6	×	✓	183	2023-10-09 07:57:02	<input type="checkbox"/>	×	▶️ ⚙️ 📄 🔗
📡 Event Publish	event	low	2	✓	✓	188	2023-10-09 07:56:25	<input type="checkbox"/>	✓	▶️ ⚙️ 📄 🔗
📄 Log After Save	log	high ?	0	×	×	185	2023-06-05 13:26:50	<input type="checkbox"/>	×	▶️ ⚙️ 📄 🔗
🔗 Object After Save	object	high ?	35	×	✓	161	2023-06-05 13:27:00	<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	2023-06-05 13:27:25	<input type="checkbox"/>	×	▶️ ⚙️ 📄 🔗

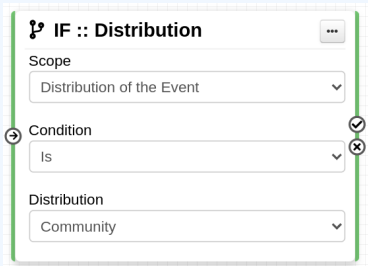
Page 1 of 1, showing 1 records out of 11 total, starting on record 1, ending on 11

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 `circl.lu`
- Or any other **generic** conditions

? These are also called **Logic modules**



The screenshot shows a configuration window for a logic module. The title is "IF :: Distribution" with a key icon on the left and a menu icon on the right. The window contains three sections, each with a dropdown menu:

- Scope:** The dropdown menu is set to "Distribution of the Event".
- Condition:** The dropdown menu is set to "Is". To the right of this section are two circular icons: a checkmark and an 'X'.
- Distribution:** The dropdown menu is set to "Community".

WORKFLOW - LOGIC MODULES

- ➡ **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 X
<input type="checkbox"/>	Module name	Type	Blocking	MISP Core format	misp-module	Custom	Enabled	Actions
<input type="checkbox"/>	🏗️ Blueprint logic module	logic	x	x	x	✓	x	▶👁️
<input type="checkbox"/>	🔄 Concurrent Task	logic	x	x	x	x	✓	■👁️
<input type="checkbox"/>	🔗 IF :: Distribution	logic	x	✓	x	x	✓	■👁️
<input type="checkbox"/>	⚙️ Filter :: Generic	logic	x	x	x	x	x	▶👁️
<input type="checkbox"/>	🗑️ Filter :: Remove filter	logic	x	x	x	x	x	▶👁️
<input type="checkbox"/>	🔗 IF :: Generic	logic	x	x	x	x	✓	■👁️
<input type="checkbox"/>	🔗 IF :: Organisation	logic	x	✓	x	x	✓	■👁️
<input type="checkbox"/>	🔗 IF :: Published	logic	x	✓	x	x	✓	■👁️
<input type="checkbox"/>	🔗 IF :: Tag	logic	x	✓	x	x	✓	■👁️
<input type="checkbox"/>	🔗 IF :: Threat Level	logic	x	x	x	x	x	▶👁️

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 'Send Mail' action module. At the top, there is a title 'Send Mail' with an envelope icon and a three-dot menu icon. Below the title is a description: 'Allow to send a Mail to a list or recipients'. The main section is titled 'Recipients' and contains a text input field with the value 'All accounts' and a close button (x). Below this is a section titled 'Mail template subject' with a text input field containing 'I'm the mail subject!'. The final section is titled 'Mail template body' with 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 input fields, indicating they can be moved or reordered.

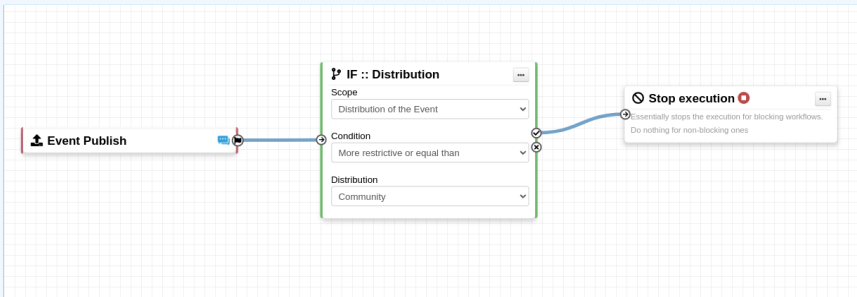
WORKFLOW - ACTION MODULES

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

All Action 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"/>	> Splunk HEC export	action	x	✓	x	x	x	▶ 🔗
<input type="checkbox"/>	🛑 Stop execution	action	✓	x	x	x	✓	■ 🔗
<input type="checkbox"/>	🏷️ Tag operation	action	x	✓	x	x	✓	■ 🔗
<input type="checkbox"/>	📌 testaction	action	x	x	✓	x	✓	■ 🔗
<input type="checkbox"/>	🔗 Webhook	action	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**



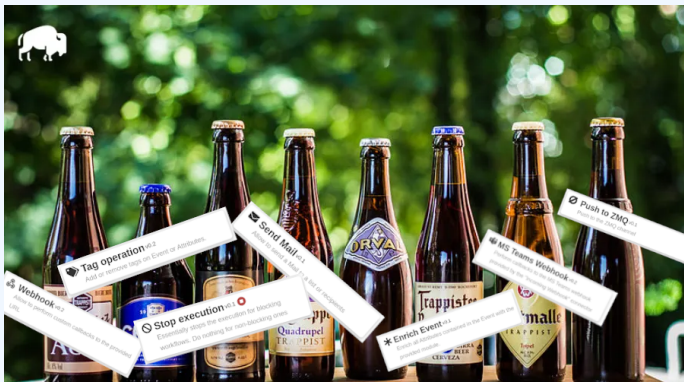
Currently 36 built-in modules.

- **Trigger** module (11): built-in **only**
 - ▶ Get in touch if you want more
- **Logic** module (10): built-in & **custom**
- **Action** module (20): 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 merge PR!)



SOURCES OF WORKFLOW MODULES (2)

User-defined **custom** modules

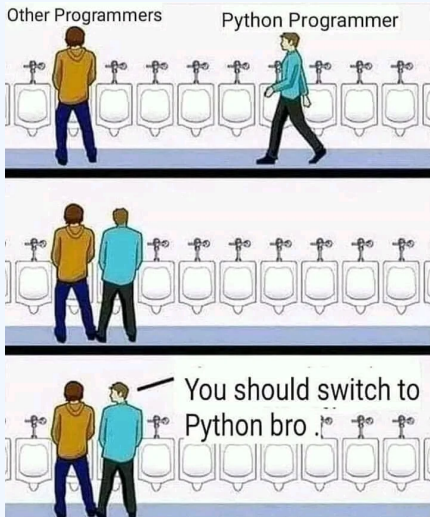
- Written in PHP
- Extend existing modules
- MISP code reuse



SOURCES OF WORKFLOW MODULES (3)

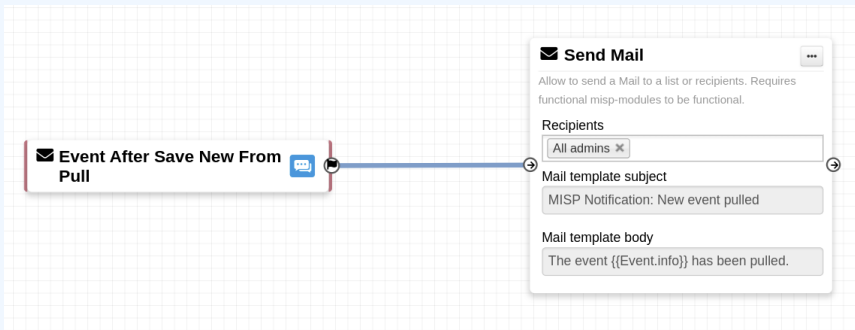
Modules from the `misp-module` **enrichment service**

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



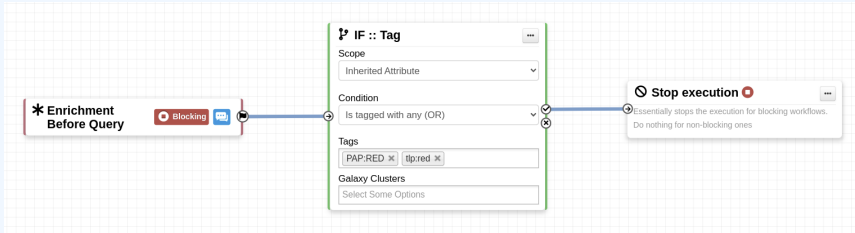
- WF-1. Send an email to **all admins** when a new event has been pulled
- WF-2. Block queries on 3rd party services when **tlp:red** or **PAP:red**
- ▶ **tlp:red**: For the eyes and ears of individual recipients only
 - ▶ **PAP:RED**: Only passive actions that are not detectable from the outside

DEMO WF-1: SEND AN EMAIL TO ALL ADMINS WHEN A NEW EVENT HAS BEEN PULLED



DEMO WF-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



Everything is ready?

Let's see how to build a workflow!



1. Prevent event publication **if tlp:red** tag
 - ▶ Send a mail to `admin@admin.test` about potential data leak
2. **else**, send a notification on Mattermost

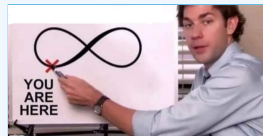
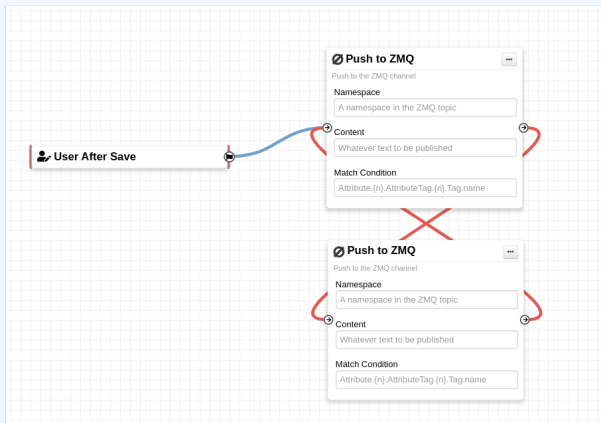
CONSIDERATIONS WHEN WORKING WITH WORKFLOWS

Objective: Overview of some common pitfalls

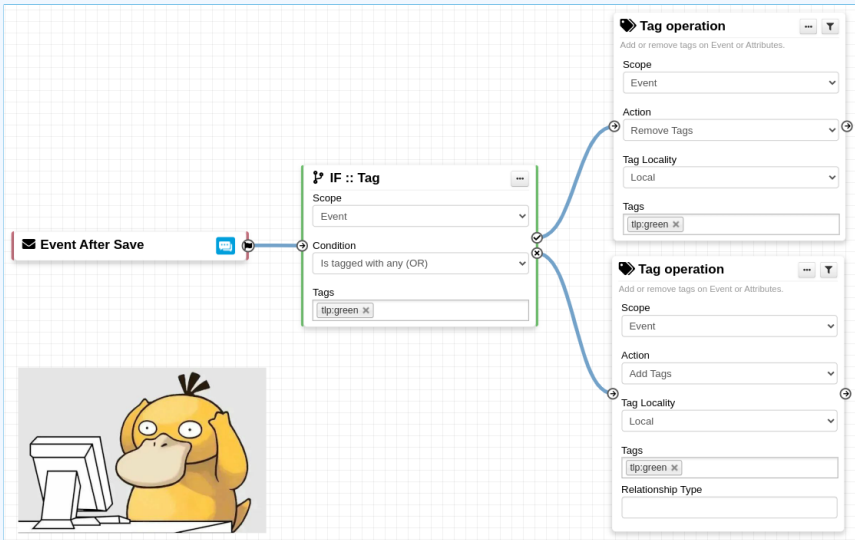


WORKING WITH THE EDITOR - OPERATIONS NOT ALLOWED

Execution loop are not authorized



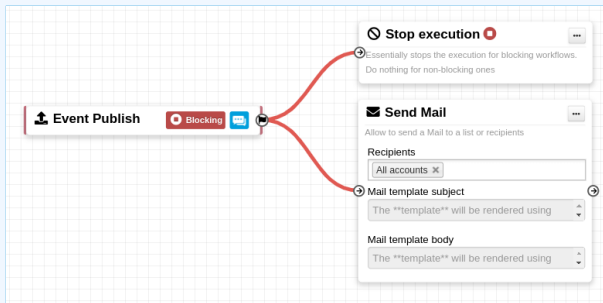
RECURSIVE WORKFLOWS



⚠ Recursion: If an action re-run the workflow

WORKING WITH THE EDITOR - OPERATIONS NOT ALLOWED

Multiple connections from the same output



- Execution order not guaranteed
- Confusing for users

ADVANCED USAGE

Objective: Overview of Blueprints, Data format and Filtering

WORKFLOW BLUEPRINTS


1. Blueprints allow to **re-use parts** of a workflow in another one
2. Blueprints can be saved, exported and **shared**

Debugging webhook v1656059209

9ff210dd-ee7e-49c8-a5af-10cd42cdadb6

Default: ✕

Blueprint Content: **1 node**

 1

Webhook module pre-configured for debugging purposes

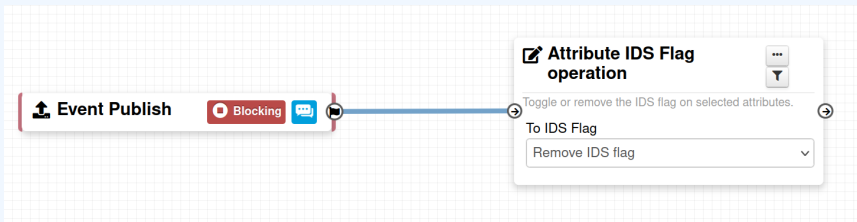
Blueprints sources: [MISP/misp-workflow-blueprints repository](https://github.com/MISP/misp-workflow-blueprints)¹

- Block actions if any attributes have the `PAP:RED` or `tlp:red` tag
- Curation pipeline
- Enrich data from 3rd-party

¹<https://github.com/MISP/misp-workflow-blueprints>

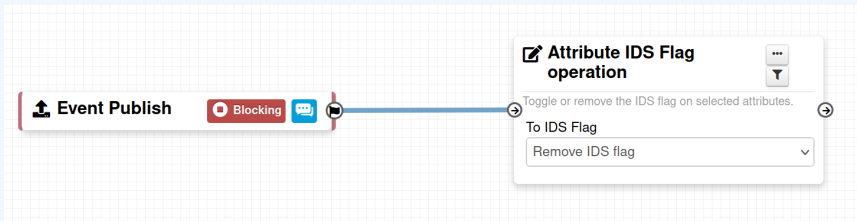
FILTERING DATA ON WHICH TO APPLY A MODULE

What is the outcome of executing this workflow?



FILTERING DATA ON WHICH TO APPLY A MODULE

What is the outcome of executing this workflow?

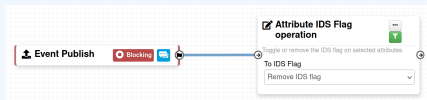


All Attributes get their `to_ids` turned off.

How could we force that action only on Attribute of type comment?

→ Hash path filtering!

FILTERING DATA ON WHICH TO APPLY A MODULE



Node Filtering

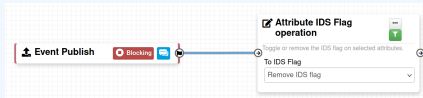
Element selector

Value

Operator

Hash Path

FILTERING DATA ON WHICH TO APPLY A MODULE



Node Filtering

Select elements on which to apply the filtering

Element selector
Event._AttributeFlattened.{n}

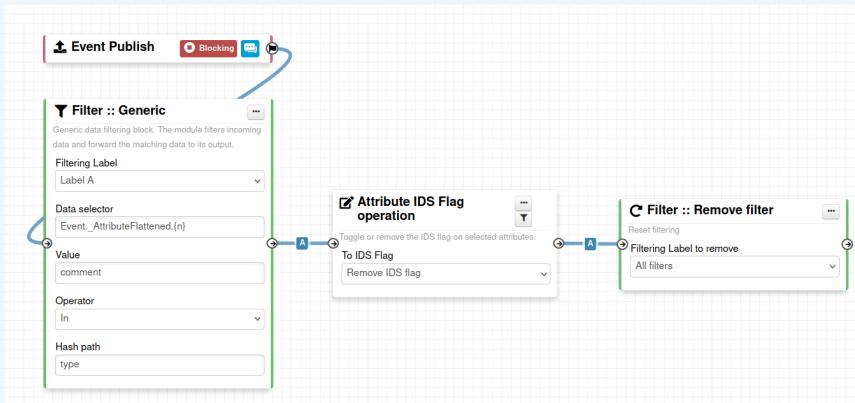
Value
comment
Fixed value

Operator
In
Comparison operator

Hash Path
type
Data point to get the value

FITLERING DATA ON WHICH TO APPLY ON MULTIPLE MODULES

New feature as of **v2.4.171** allows setting filters on a path.



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

- I have a curation pipeline using the API, should I port it to workflows?
 - ▶ **No** in general, but WF can be used to start the curation process or perform simple pre-processing
- What if I want to **block** some actions
 - ▶ Put the blocking logic in the WF, keep the remaining outside
- Bottom line is **Keep it simple** for you to maintain

FUTURE WORKS

- More 🎬 modules
- More ➡ modules
- More 🐘 triggers
- Recursion prevention system



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

