MISP

INTELLIGENCE SHARING

Automation with Workflows in MISP

Short version

Sami Mokaddem

MISP Project

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

Content of the presentation

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

Automation in MISP: What already exists?

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

Simple automation in MISP made easy

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

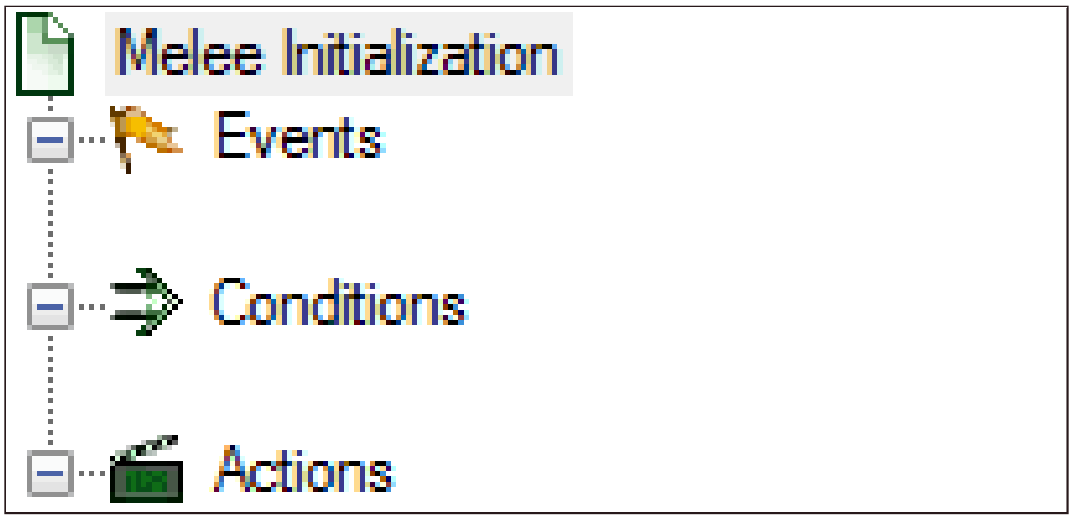
Example of use-cases

- **Notification** on specifc 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

What kind of events?

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	☐	✗	
Enrichment Before Query	others	low	2226	✓	✓	162	2023-10-09 07:56:42	☐	✓	
Event After Save	event	high	191	✗	✓	175	2023-10-02 14:55:19	☐	✗	
Event After Save New	event	low	7	✗	✓	182	2023-03-16 14:05:07	☐	✗	
Event After Save New From Pull	event	low	6	✗	✓	183	2023-10-09 07:57:02	☐	✗	
Event Publish	event	low	2	✓	✓	188	2023-10-09 07:56:25	☐	✓	
Log After Save	log	high	0	✗	✗	185	2023-06-05 13:26:50	☐	✗	
Object After Save	object	high	35	✗	✓	161	2023-06-05 13:27:00	☐	✗	
Post After Save	post	low	36	✗	✗	176	2022-07-28 13:59:51	☐	✗	
User After Save	user	low	0	✗	✗	181	2022-08-05 07:19:46	☐	✗	
User Before Save	user	low	42	✓	✗	158	2023-06-05 13:27:25	☐	✗	

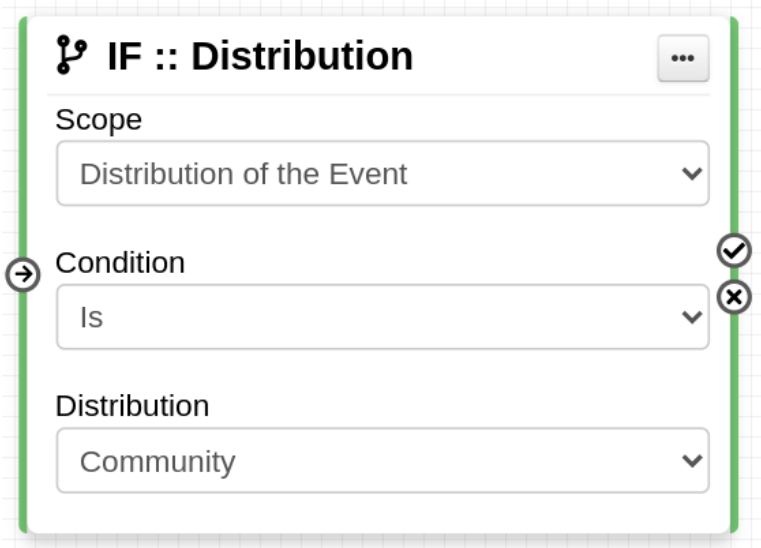
Page 1 of 1, showing 1 records out of 11 total, starting on record 1, ending on 11

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**



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

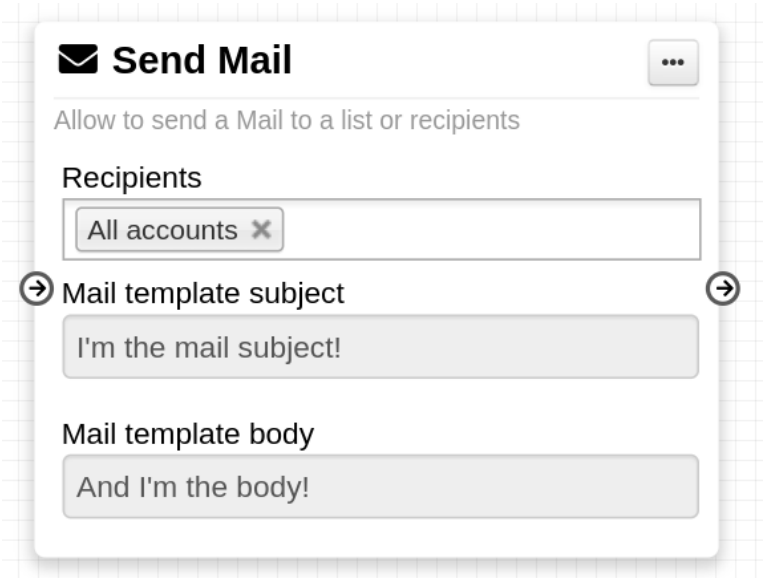
Module name	Type	Blocking	MISP Core format	misp-module	Custom	Enabled	Actions
Blueprint logic module	logic	✗	✗	✗	✓	✗	
Concurrent Task	logic	✗	✗	✗	✗	✓	
IF :: Distribution	logic	✗	✓	✗	✗	✓	
Filter :: Generic	logic	✗	✗	✗	✗	✗	
Filter :: Remove filter	logic	✗	✗	✗	✗	✗	
IF :: Generic	logic	✗	✗	✗	✗	✓	
IF :: Organisation	logic	✗	✓	✗	✗	✓	
IF :: Published	logic	✗	✓	✗	✗	✓	
IF :: Tag	logic	✗	✓	✗	✗	✓	
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**



Workflow - Action modules

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

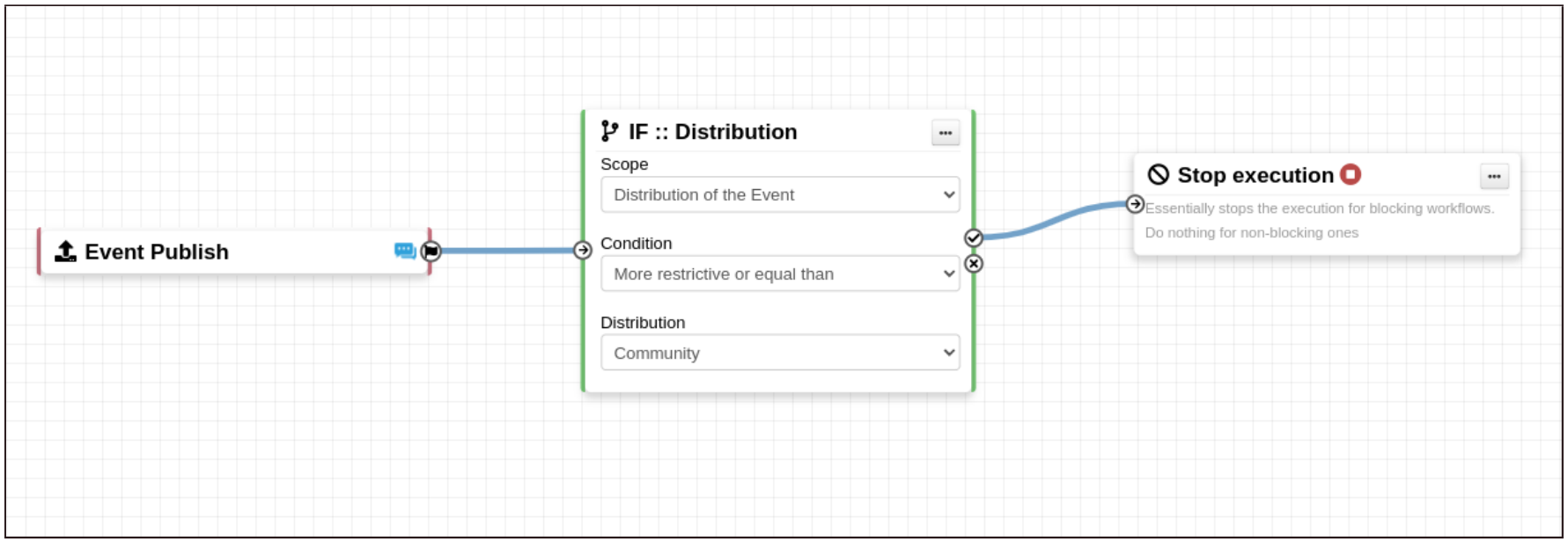
All | Action | Logic | misp-module | Custom | Blocking | Enabled | Disabled

Enter value to search | Filter

Module name	Type	Blocking	MISP Core format	misp-module	Custom	Enabled	Actions
Attach enrichment	action	✗	✓	✗	✗	✓	
Attribute edition operation	action	✗	✓	✗	✗	✓	
Attribute IDS Flag operation	action	✗	✓	✗	✗	✓	
Blueprint action module	action	✗	✗	✗	✓	✓	
Enrich Event	action	✗	✓	✗	✗	✓	
mattermost	action	✗	✗	✓	✗	✓	
MS Teams Webhook	action	✗	✗	✗	✗	✓	
Push to ZMQ	action	✗	✗	✗	✗	✓	
Send Log Mail	action	✗	✗	✗	✗	✗	
Send Mail	action	✗	✗	✗	✗	✓	
Splunk HEC export	action	✗	✓	✗	✗	✗	
Stop execution	action	✓	✗	✗	✗	✓	
Tag operation	action	✗	✓	✗	✗	✓	
testaction	action	✗	✗	✓	✗	✓	
Webhook	action	✗	✗	✗	✗	✓	

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**



Sources of Workflow modules (0)

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)

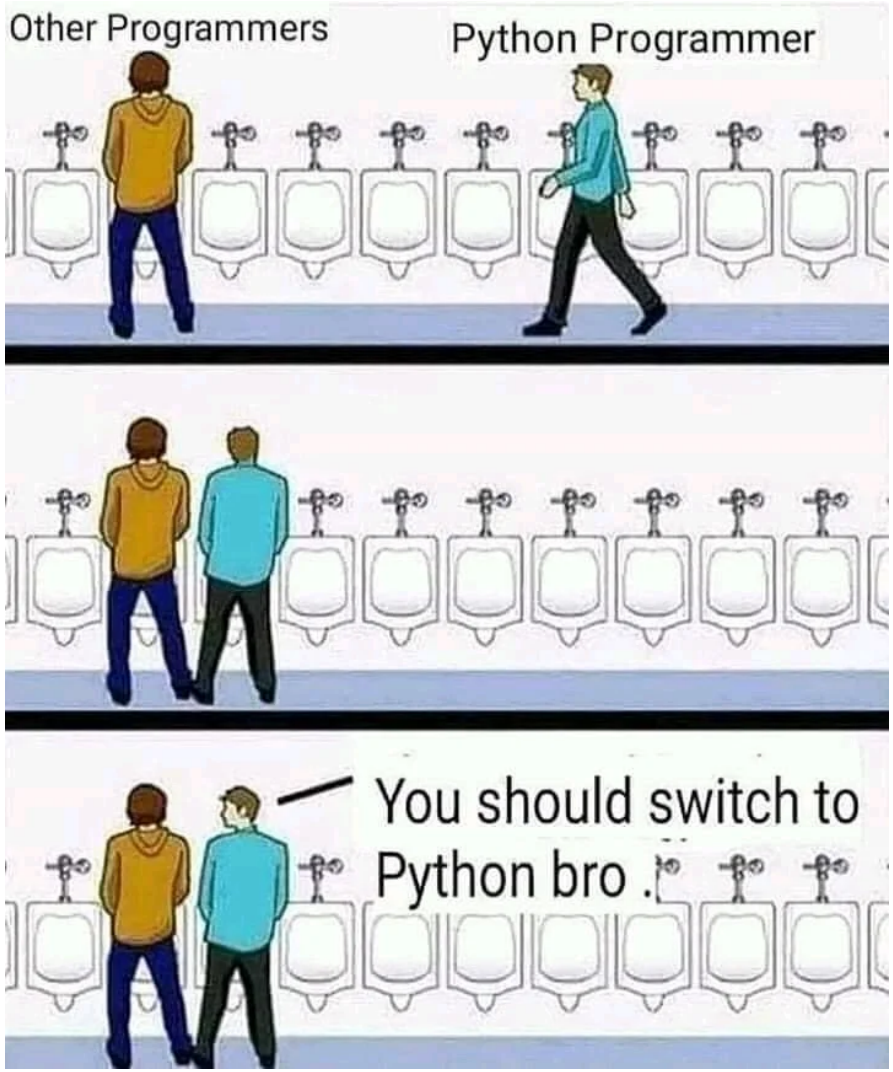
User-defined **custom** modules

- 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



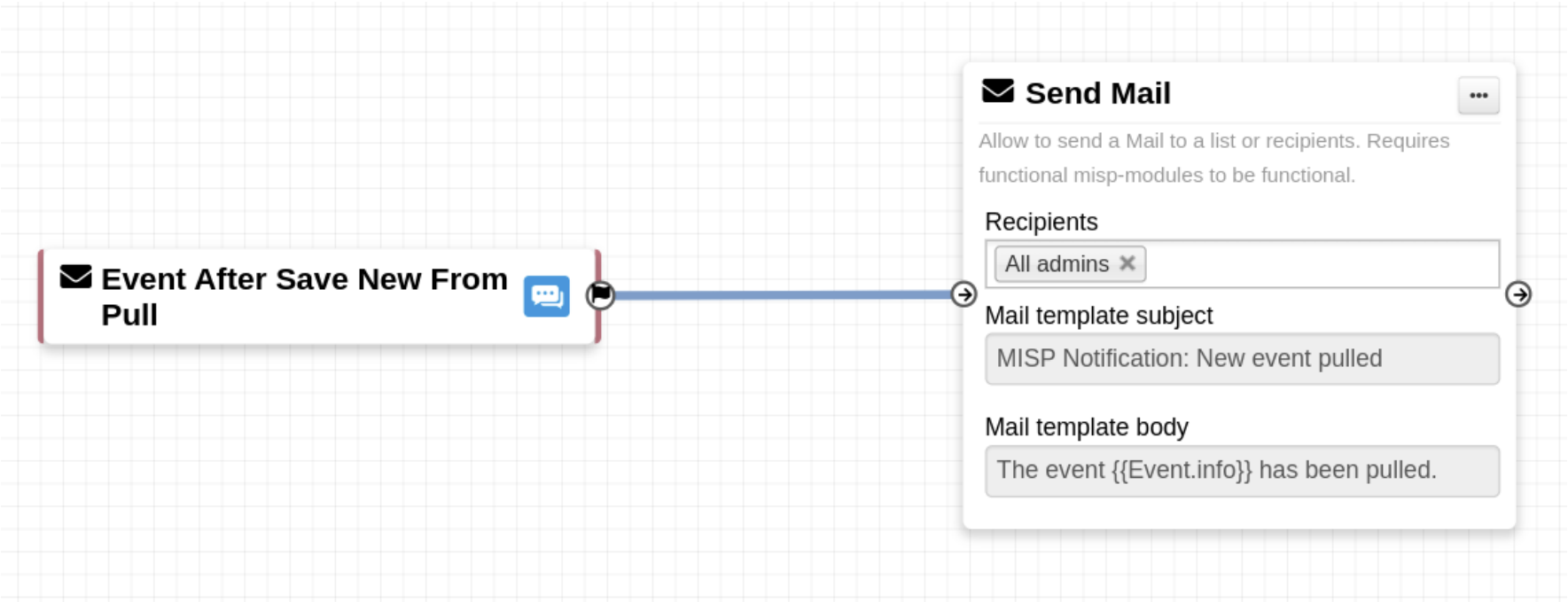
Demo by examples

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!

Creating a workflow with the editor

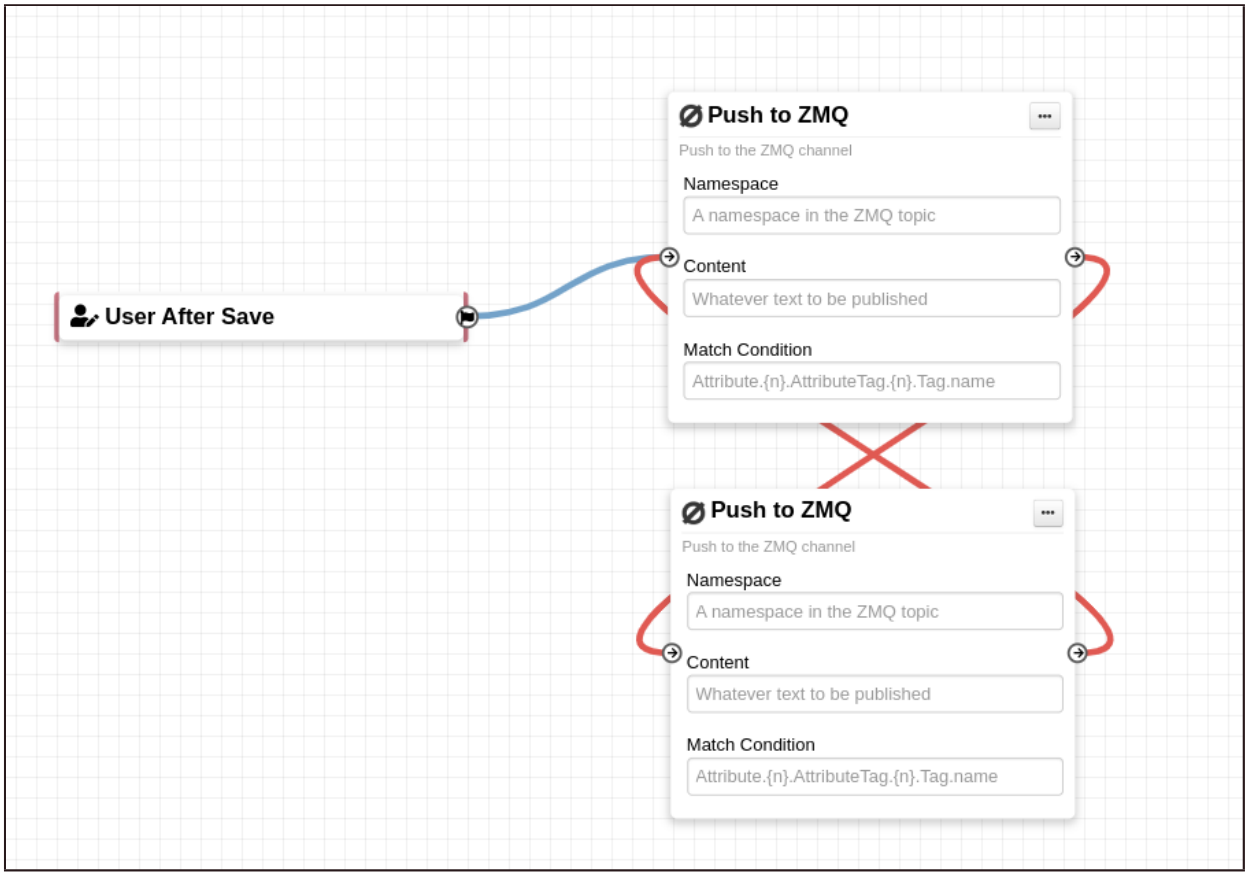
1. Prevent event publication **if tlp:red** tag
   - Send a mail to `admin@admin.test` about potential data leak
2. **else**, send a notification on Mattermost

Considerations when working with workflows

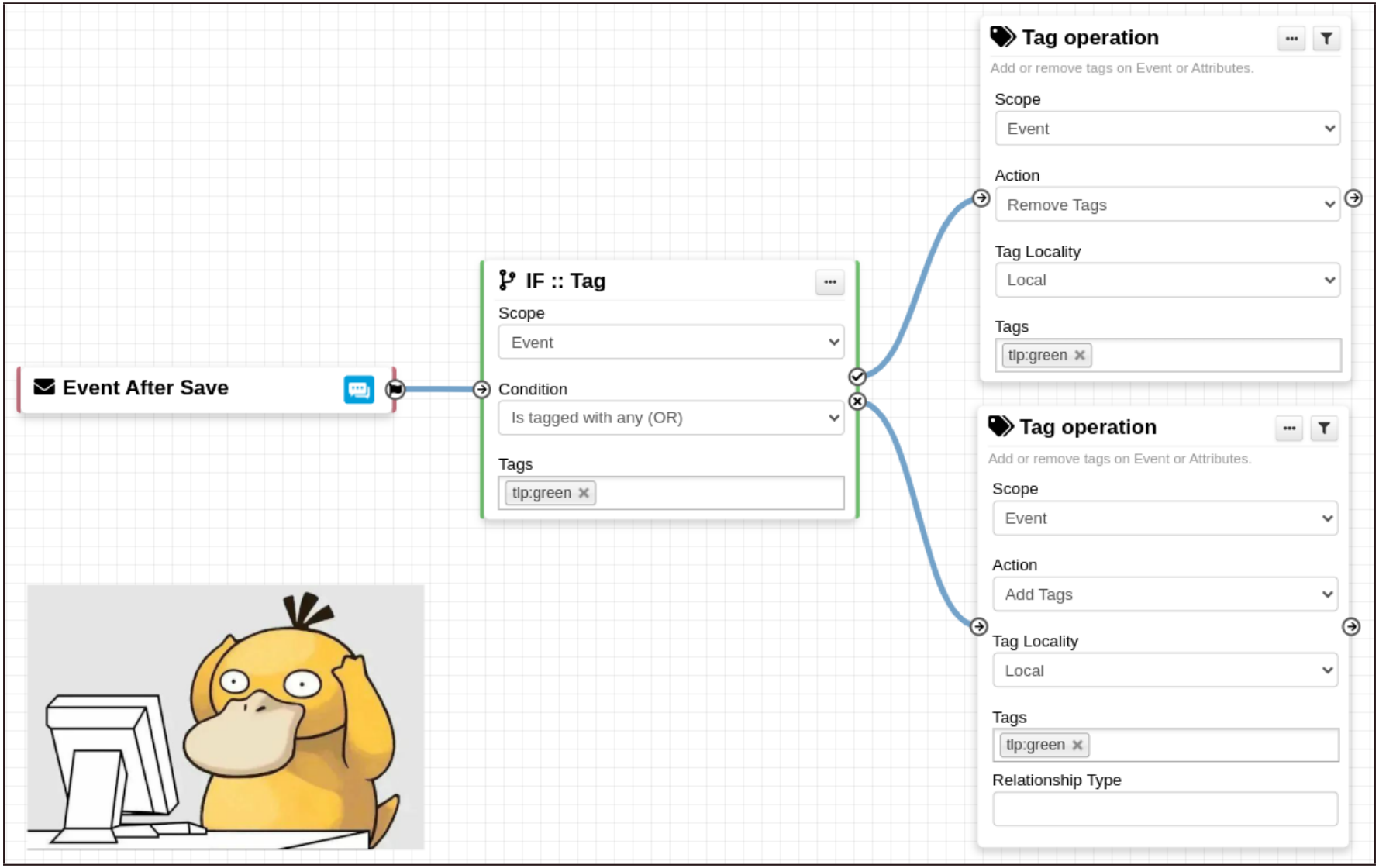
Objective: Overview of some common pitfalls

Working with the editor - Operations not allowed

Execution loop are not authorized



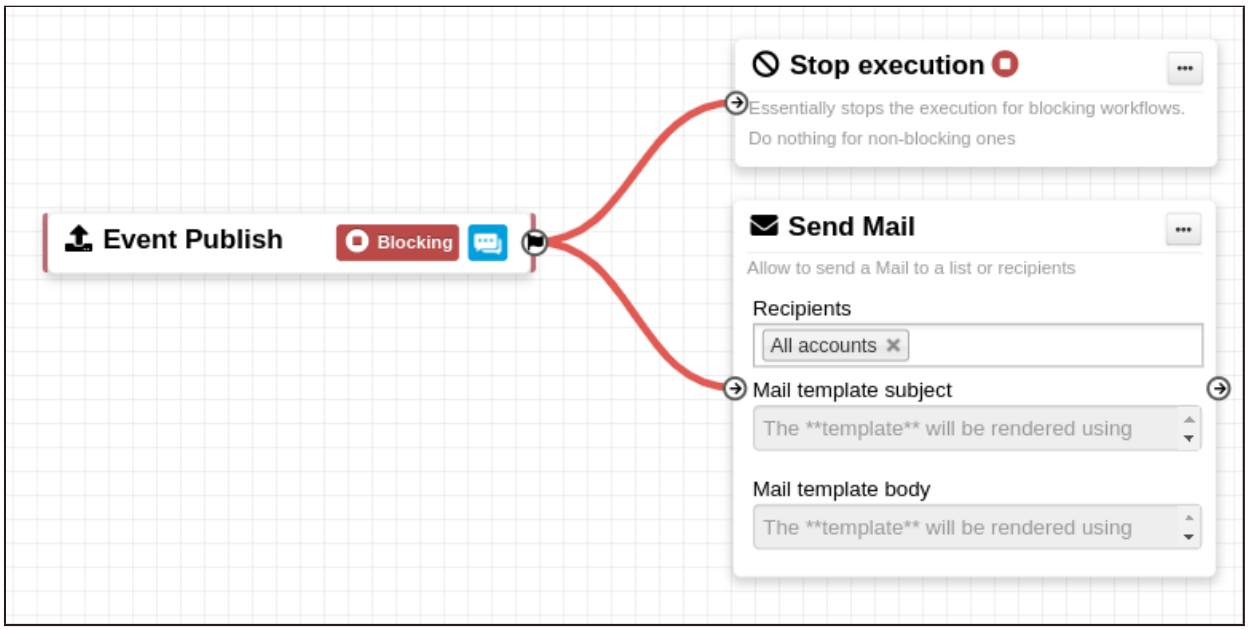
Recursive workflows



⚠Recursion: If an action re-run the workflow

Working with the editor - Operations not allowed

Multiple connections from the same output



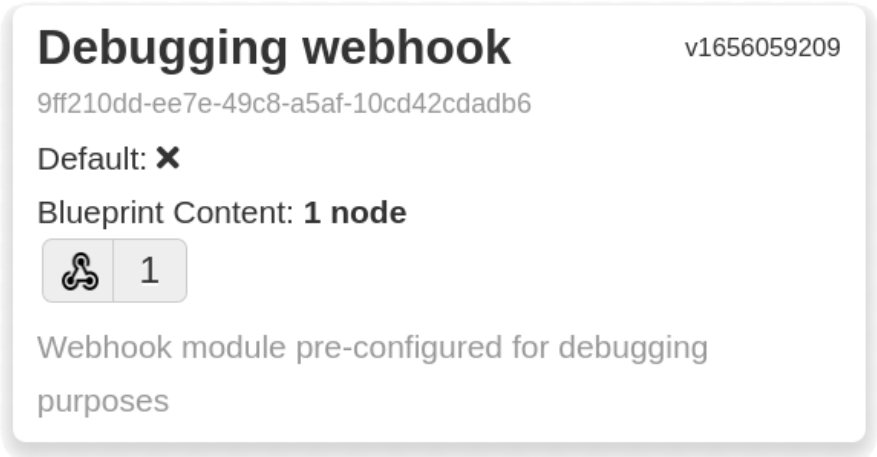
- Execution order not guaranted
- Confusing for users

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**



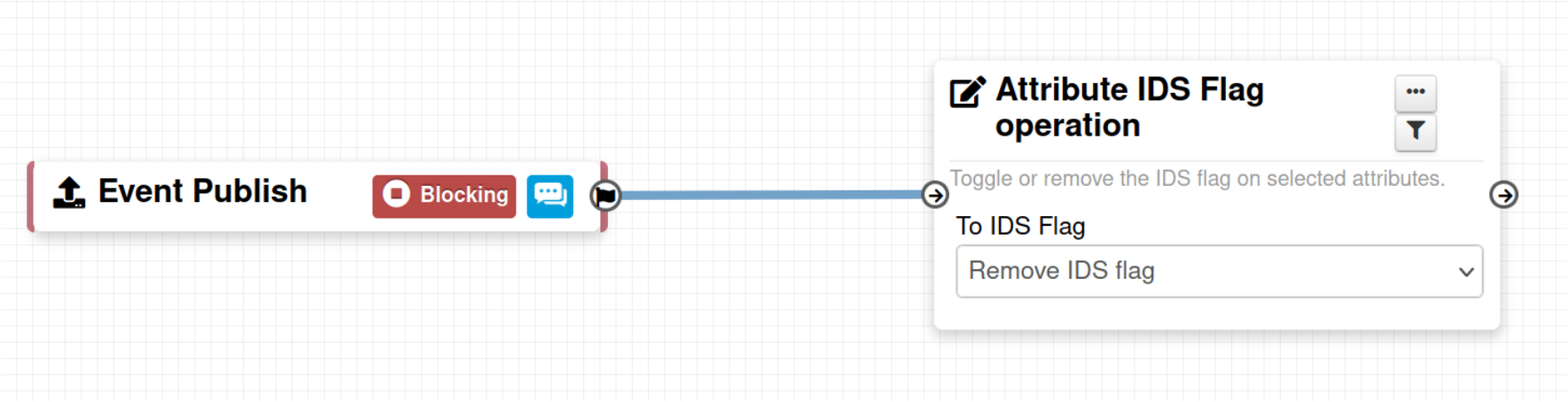
Blueprints sources: `MISP/misp-workflow-blueprints` repository[1]

- 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

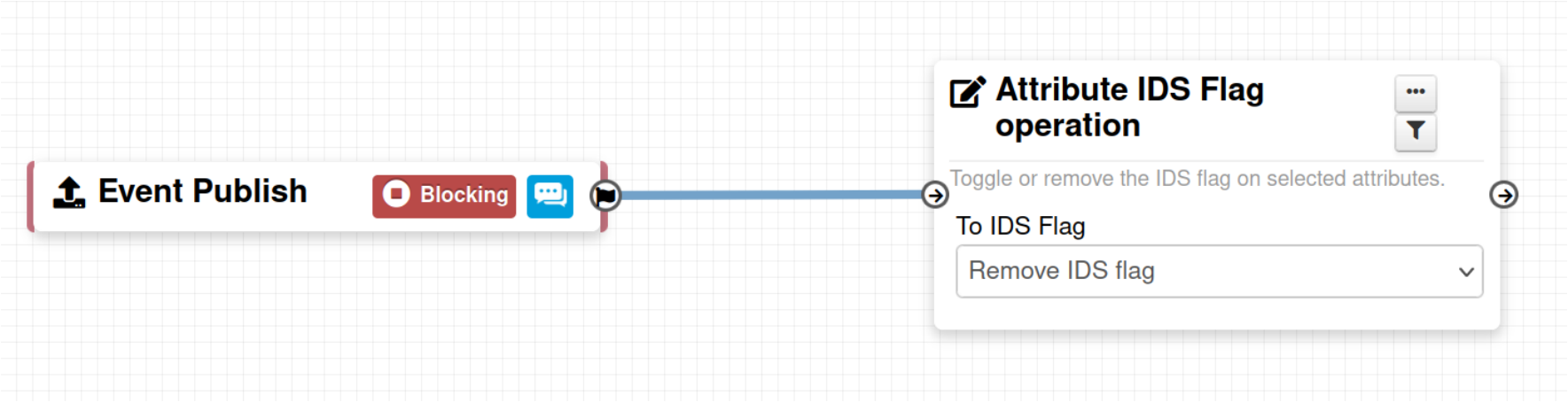
Fitlering data on which to apply a module

What is the outcome of executing this workflow?



Fitlering data on which to apply a module

What is the outcome of executing this workflow?

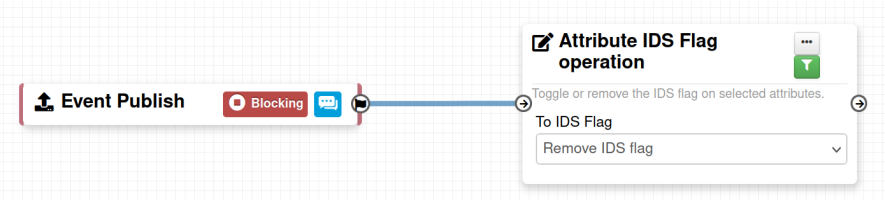


All Attributes get their `to_ids` turned off.

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

→ Hash path filtering!

Fitlering data on which to apply a module



Node Filtering

Element selector

Event._AttributeFlattened.{n}

Value

comment

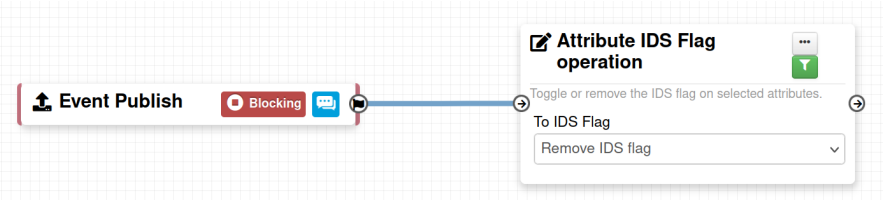
Operator

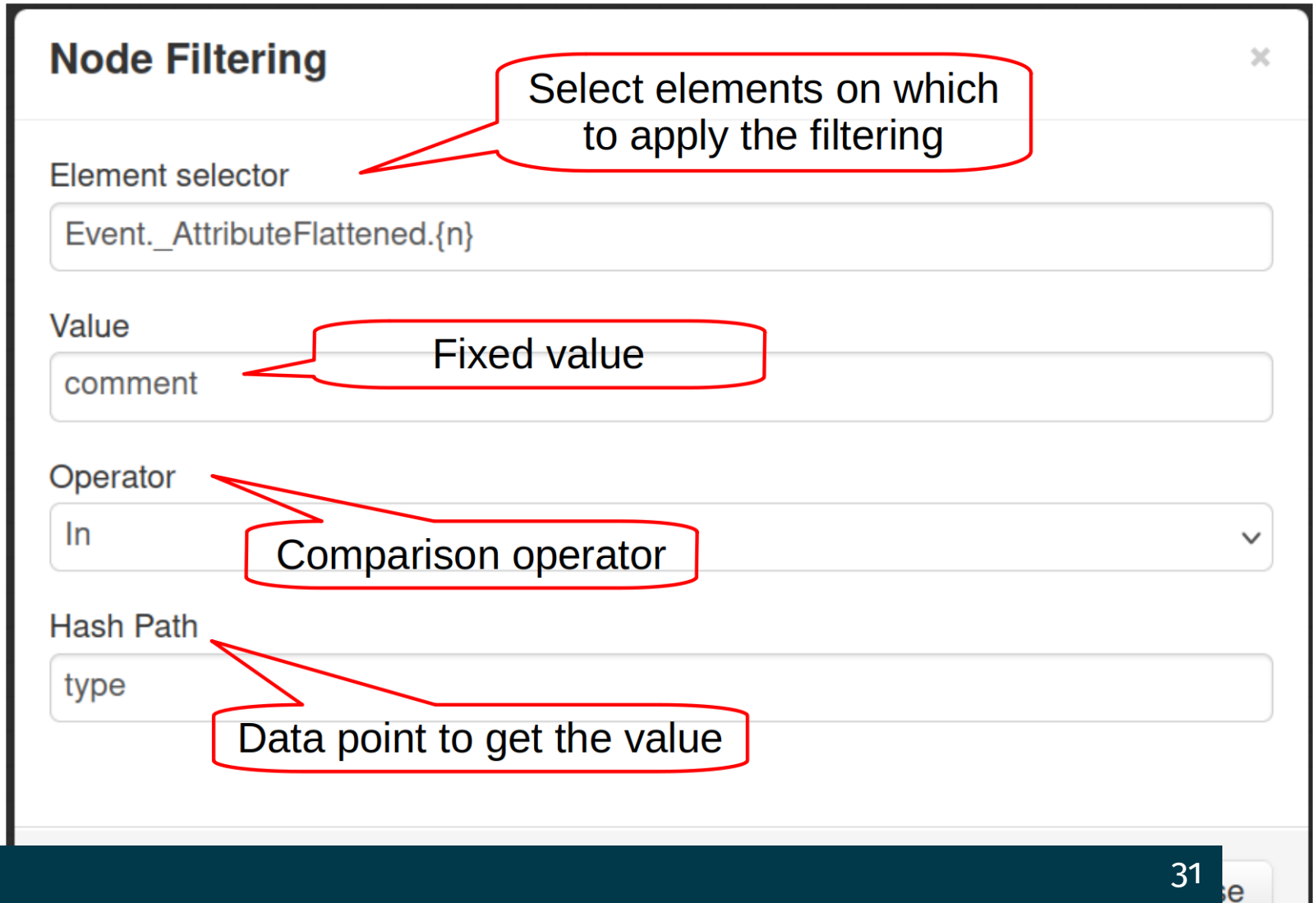
In

Hash Path

type

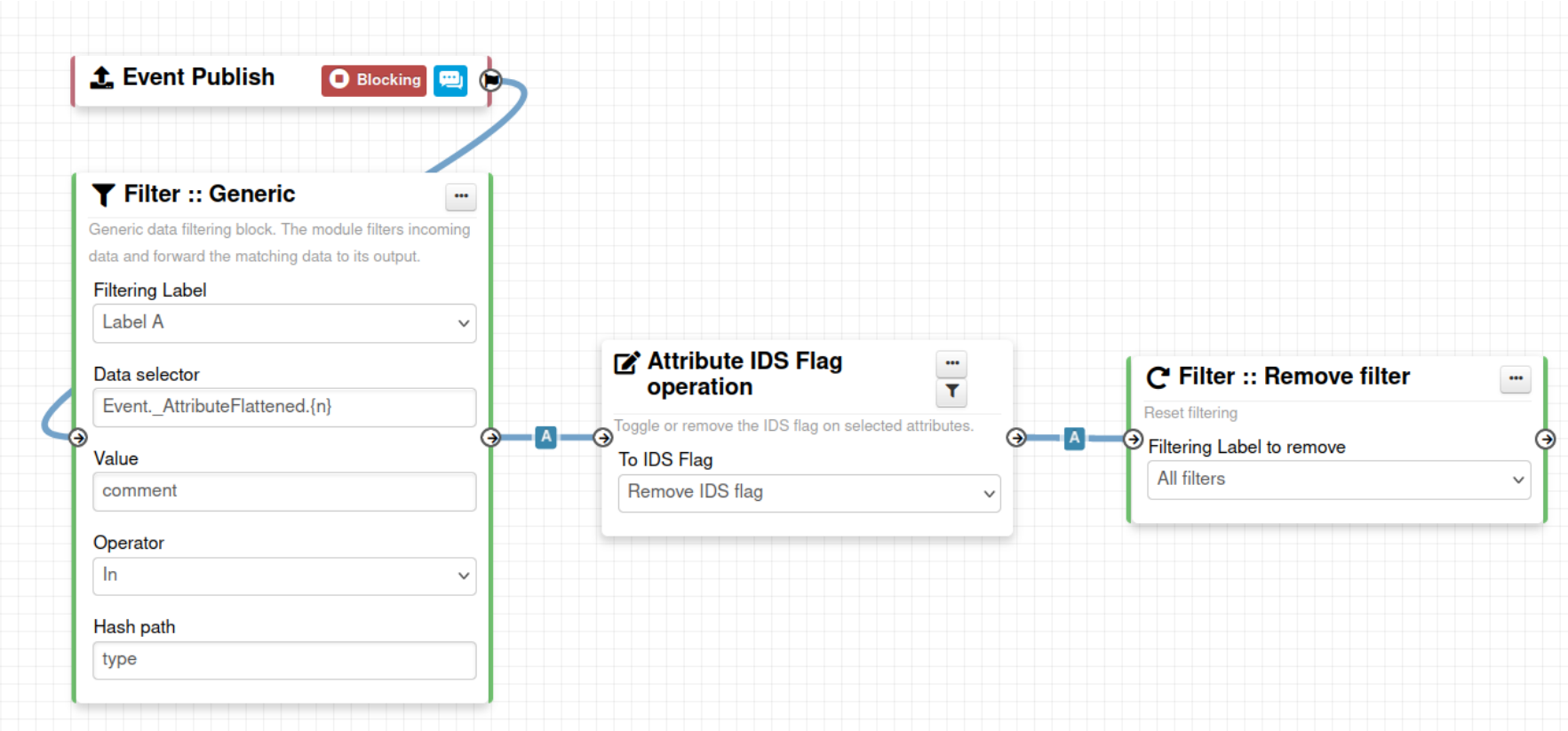
Fitlering data on which to apply a module





Fitlering data on which to apply on multiple modules

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



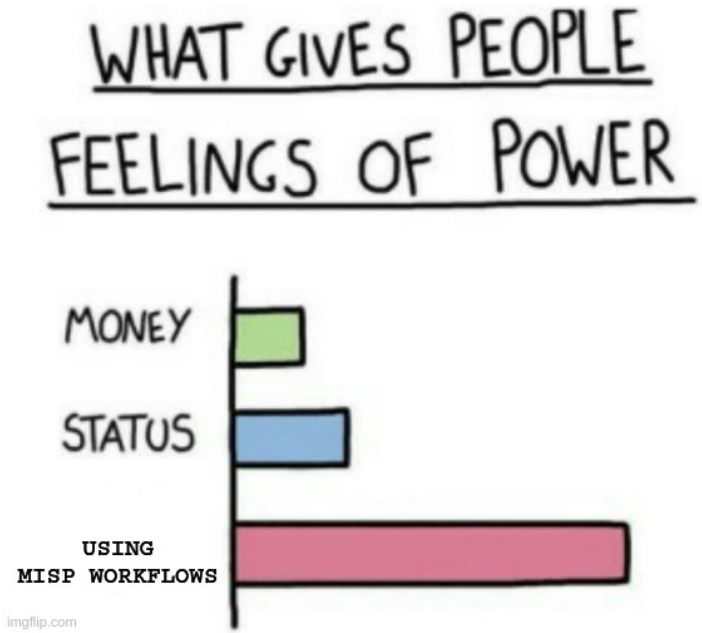
Should I migrate to MISP Workflows

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

Final words

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