

# Automation with Workflows in MISP

Advanced version

Sami Mokaddem

MISP Project

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



# EXAMPLE OF USE-CASES

- **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



# TRIGGERS

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](#)

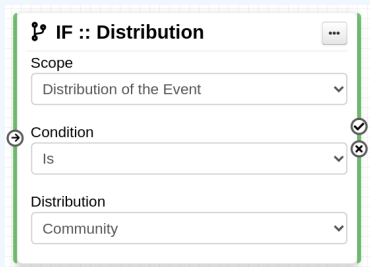
« previous next »

| 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

⇒ **Conditions** ? **Logic modules** allow to redirect the execution flow

- 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



The screenshot shows a configuration window for a logic module titled "IF :: Distribution". The window has a yellow header bar with a key icon and a title. Below the header, there are three sections, each with a label and a dropdown menu. The first section is labeled "Scope" and has a dropdown menu with "Distribution of the Event" selected. The second section is labeled "Condition" and has a dropdown menu with "Is" selected. The third section is labeled "Distribution" and has a dropdown menu with "Community" selected. On the right side of the "Condition" section, there are two circular icons: a checkmark and an 'X'. On the left side of the "Condition" section, there is a circular icon with a right-pointing arrow. The window has a green border and a close button in the top right corner.



Actions



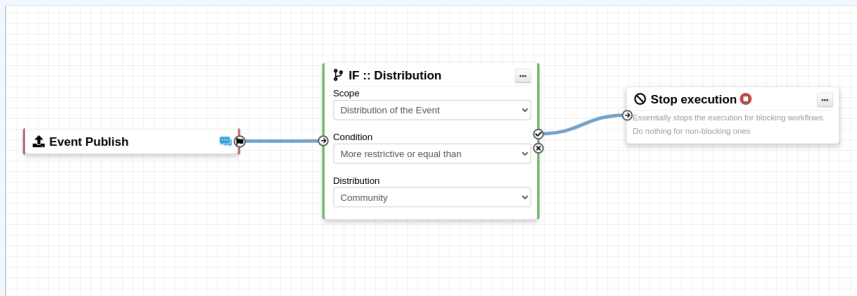
**Action modules** allow to executes operations

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

The screenshot shows a configuration window for the 'Send Mail' action. It has a title bar with an envelope icon and the text 'Send Mail'. Below the title, it says 'Allow to send a Mail to a list or recipients'. There are three main sections: 'Recipients' with a dropdown menu showing 'All accounts' and a close button; 'Mail template subject' with a text input field containing 'I'm the mail subject!'; and 'Mail template body' with a text input field containing 'And I'm the body!'. Navigation arrows are visible on the left and right sides of the configuration area.

# 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

## Built-in **default** modules

- Part of the MISP codebase
- Ready to use once enabled

## User-defined **custom** modules

### Written in PHP

- Extend existing modules
- MISP code reuse

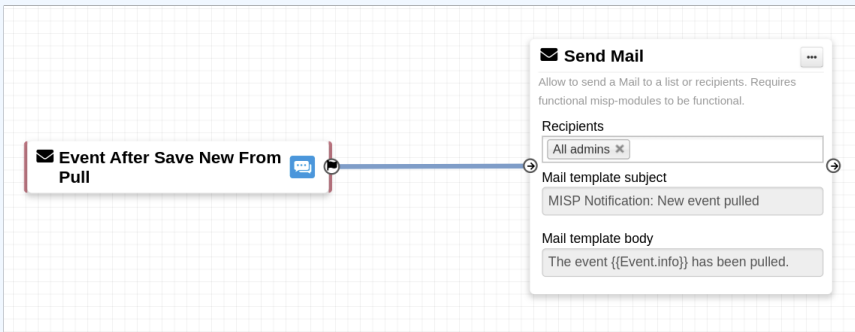
### Written in Python

- Can rely on extensive python libraries
- Easier to write
- Rely on the **enrichment service** 



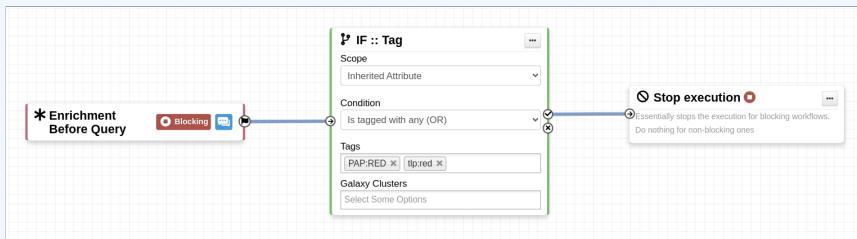
- WF-1. Send an email to **all admins** when a new event has been pulled
- WF-2. Block queries on 3rd party services when **tlp:red** or **PAP:red**
- ▶ **tlp:red**: For the eyes and ears of individual recipients only
  - ▶ **PAP:RED**: Only passive actions that are not detectable from the outside

# 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



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

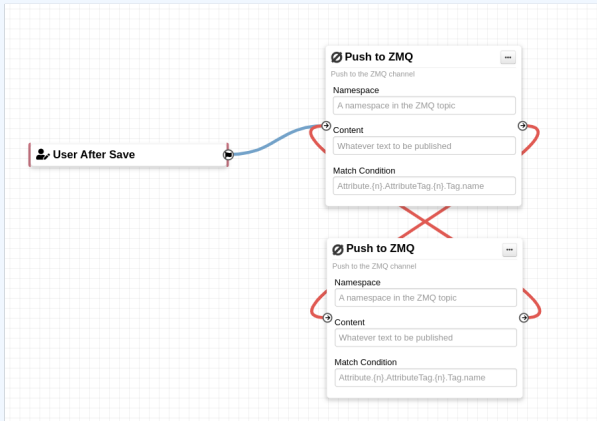
# CONSIDERATIONS WHEN WORKING WITH WORKFLOWS

**Objective:** Overview of some common pitfalls

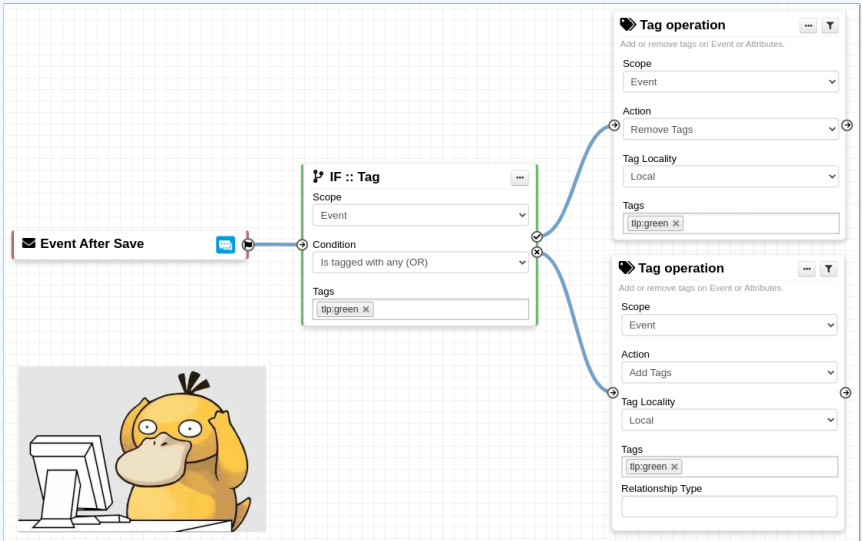


# WORKING WITH THE EDITOR - OPERATIONS NOT ALLOWED

Execution loop are not authorized



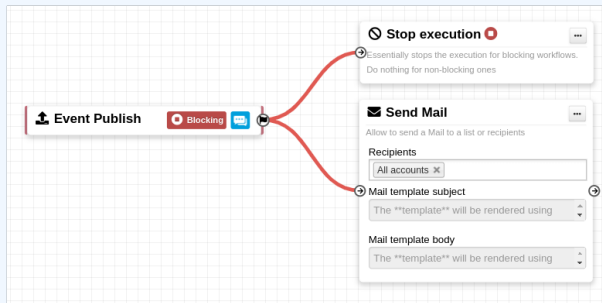
# RECURSIVE WORKFLOWS



⚠ 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



# NEW RECENT FEATURES

## ■ New action modules & improvements

- ▶ Assign country
- ▶ Attach warninglist
- ▶ Attribute operations
- ▶ Tag replacements
- ▶ Webhook, ...

## ■ New logic modules & improvements

- ▶ Filter :: Generic
- ▶ Filter :: Remove
- ▶ IF :: \*

# NEW RECENT FEATURES I

**Assign country**  
Add or remove country Galaxy Cluster based on provided data  
Scope: Event  
Country Hash path: enrichment.[n].[n].values.0  
Tag Locality: Local  
Galaxy Name: country

**Add to warninglist**  
Append attributes to an active custom warninglist.  
Warninglist: Confirmed false-positive

**Event distribution operation**  
Set the Event's distribution to the selected level  
Preserve timestamp and published state  
Modify timestamp and unpublish  
Distribution: Organisation

**Add Event Blocklist entry**  
Create a new entry in the Event blocklist table  
Event UUID Hash path: Event.uuid  
Event Info Hash path: Event.info  
Blocklist Comment: Blocked from workflow

**Attach warninglist**  
Attach selected warninglist result.  
Warninglists: [ALL x]

**Tag Replacement - TLP**  
Attach a tag (or substitute) a tag by another for the TLP taxonomy  
Scope: Event  
Removed substituted tag: No  
Tag Locality: Local

**Tag Replacement - PAP**  
Attach a tag (or substitute) a tag by another for the PAP taxonomy  
Scope: Event  
Removed substituted tag: No  
Tag Locality: Local

**Attribute IDS Flag operation**  
Toggle or remove the IDS flag on selected attributes.  
To IDS Flag: Toggle IDS flag

**Webhook**  
Allow to perform custom callbacks to the provided URL  
URL: https://example.com/test  
Content type: application/json  
HTTP Request Method: POST  
Self-signed certificates: Deny self-signed certificates  
Payload (leave empty for roaming data):  
Headers: Authorization: foobar

**Attribute comment operation**  
Set the Attribute's comment to the selected value  
Comment: Comment to be set

**IF :: Published**  
Condition: Event is published

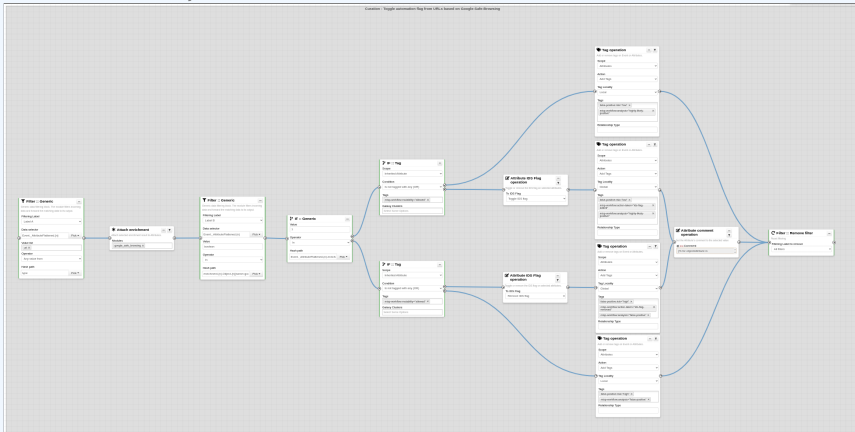
**IF :: Count**  
Data selector to count: Event.Tag [n].name  
Condition: Equals to  
Value: 50

**Filter :: Remove filter**  
Reset filtering  
Filtering Label to remove: All filters

**Filter :: Generic**  
Generic data filtering block. The module filters incoming data and forward the matching data to its output.  
Filtering Label: Label A  
Data selector: Event.\_AttributeFlattened [n]  
Value: tlp:red  
Operator: In  
Hash path: Tag.name

# NEW RECENT FEATURES II

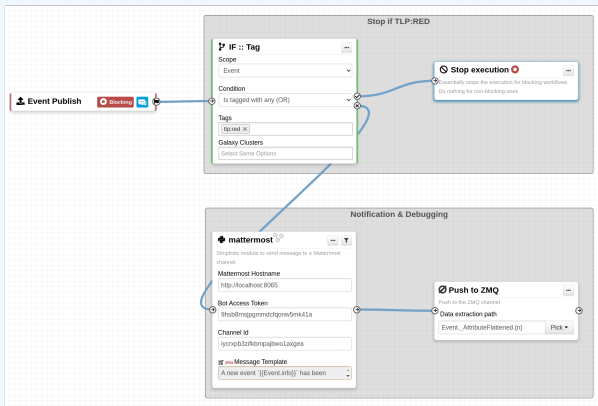
## ~ 12 New blueprints for IoC curation



# NEW RECENT FEATURES III

## ■ UI improvements

- ▶ Frame to annotate and group modules
- ▶ More documentation (Format, Jinja2 syntax)
- ▶ Collapsible sidebar and quick node insert
- ▶ Hash path picker



# ADVANCED USAGE

# ADVANCED USAGE

## Objective:

- Blocking workflows
- Blueprints
- Filtering
- Data format
- Debugging

Two types of workflows:

## **Blocking** Workflows



- ▶ Can prevent / block the original event to happen
- ▶ If a **blocking module**  blocks the action
- ▶ event-publish, event-before-save, enrichment-before-query, ...

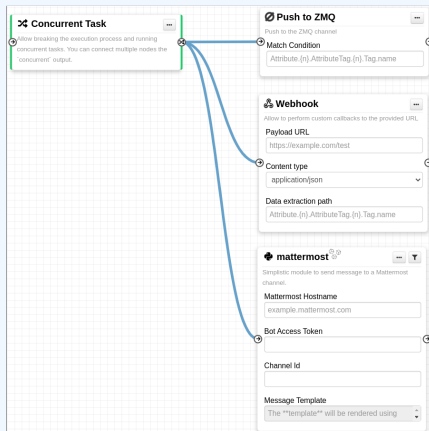
## **Non blocking** Workflows execution outcome has no impact

- ▶ No way to prevent something that happened in the past
- ▶ event-after-save, attribute-after-save log-after-save, ...



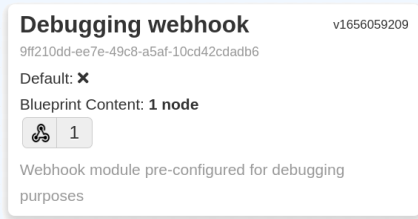
# LOGIC MODULE: CONCURRENT TASK

- Logic module allowing **multiple output** connections
- **Postpone the execution** for remaining modules
- Convert  **Blocking** →  **Non blocking**



# 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<sup>1</sup>

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

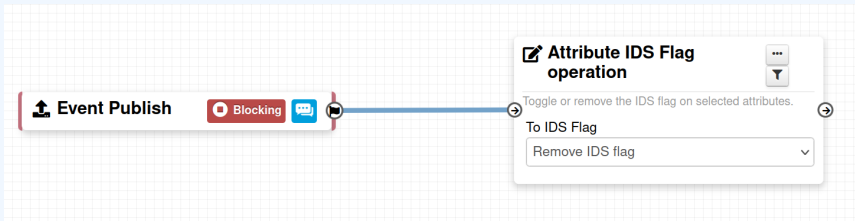
---

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

# FILTERING

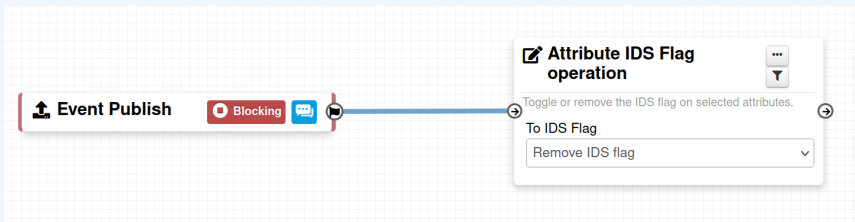
# FILTERING DATA ON WHICH TO APPLY A MODULE

What is the outcome of executing this workflow?



# FILTERING DATA ON WHICH TO APPLY A MODULE

What is the outcome of executing this workflow?

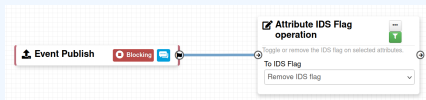


All Attributes get their `to_ids` turned off.

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

Event.\_AttributeFlattened.{n}

Value

comment

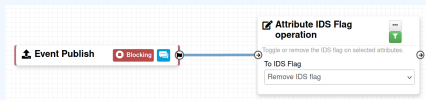
Operator

In

Hash Path

type

# 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

# FILTERING DATA ON WHICH TO APPLY A MODULE

```
{
  "Event": {
    "id": "64",
    "org_id": "1",
    "org_id": "1",
    "date": "2023-05-03",
    "info": "Core format sample",
    "published": false,
    "uuid": "b9957473-bb46-4c65-b69e-974b3c93c1f4",
    "analysis": "Q",
    "timestamp": "1683117902",
    "Attribute": [
      {
        "id": "1695",
        "type": "ip-src",
        "category": "Network activity",
        "to_ids": true,
        "uuid": "9ac36927-0874-4094-bf4c-f922c1e9cc35",
        "event_id": "64",
        "distribution": "5",
        "timestamp": "1683117902",
        "comment": "",
        "sharing_group_id": "0",
        "deleted": false,
        "disable_correlation": false,
        "object_id": "0",
        "object_relation": null,
        "first_seen": null,
        "last_seen": null,
        "value": "8.8.8.8",
        "Galaxy": [],
        "ShadowAttribute": [],
        "Tag": [
          {
            "id": "137",
            "name": "PAP:AMBER",
            "colour": "#ffa800",
```

Extract elements  
on which to apply  
the filter

{**"id"**: 1, **"value"**: "8.8.8.8", **"type"**: "ip-src"}

{**"id"**: 2, **"value"**: "8.8.4.4", **"type"**: "ip-dst"}

{**"id"**: 3, **"value"**: "circl.lu", **"type"**: "domain"}

{**"id"**: 4, **"value"**: "-- test --", **"type"**: "comment"}

For each  
element

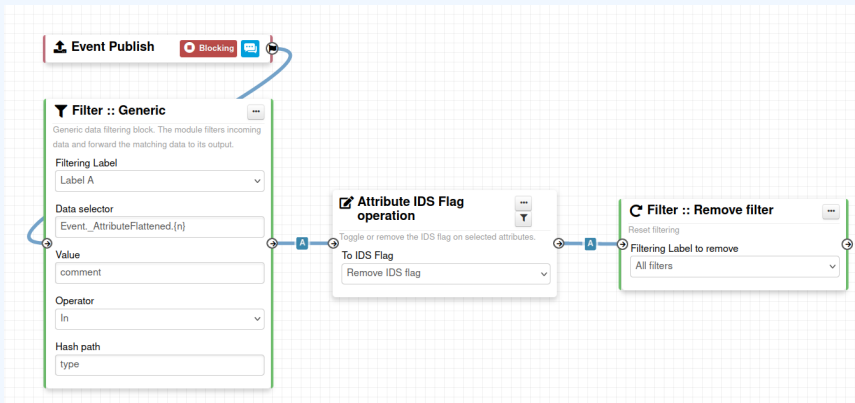
1. Access the entry's value from the hash path
2. Compare two values using the operator
3. Discard invalid elements

{**"id"**: 4, **"value"**: "-- test --", **"type"**: "comment"}



# FILTERING DATA ON WHICH TO APPLY ON MULTIPLE MODULES

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



# DATA FORMAT IN WORKFLOWS

- In most cases, the format is the **MISP Core format**
  - ▶ Attributes are **always encapsulated** in the Event or Object

```
{
  ▾ Event : { 31 }
    id : 64
    orgc_id : 1
    org_id : 1
    date : 2023-05-03
    threat_level_id : 1
    Info : Core format sample
    published : false
    uuid : b9557473-bb46-4c65-b69e-974b3c93c1f4
    attribute_count : 2
    analysis : 0
    timestamp : 1683117902
    distribution : 1
    proposal_email_lock : false
    locked : false
    publish_timestamp : 0
    sharing_group_id : 0
    disable_correlation : false
    extends_uuid :
    protected : null
    event_creator_email : admin@admin.test
    ▶ Org : { 4 }
    ▶ Orgc : { 4 }
    ▶ Attribute : { 1 }
    ▶ ShadowAttribute : [ 0 ]
    ▶ RelatedEvent : [ 0 ]
    ▶ Galaxy : [ 1 ]
    ▶ Object : [ 1 ]
    ▶ EventReport : [ 0 ]
    ▶ CryptographicKey : [ 0 ]
    ▶ Tag : [ 2 ]
  }
```

# HASH PATH FILTERING - EXAMPLE

```
1 {  
2   "Event": {  
3     "uuid": ...  
4     "timestamp": ...  
5     "distribution": 1,  
6     "published": false,  
7     "Attribute": [  
8       {  
9         "type": "ip-src",  
10        "value": "8.8.8.8", ...  
11      },  
12      {  
13        "type": "domain",  
14        "value": "misp-project.org", ...  
15      }  
16    ],  
17    ...  
18  }  
19 }
```

## 1. Access Event distribution

- ▶ Event.distribution

# HASH PATH FILTERING - EXERCISE (1)

```
1 {  
2   "Event": {  
3     "uuid": ...  
4     "distribution": 1,  
5     "published": false,  
6     "Attribute": [  
7       {  
8         "type": "ip-src",  
9         "value": "8.8.8.8", ...  
10      },  
11      {  
12        "type": "domain",  
13        "value": "misp-project.org", ...  
14      }  
15    ],  
16    ...  
17  }  
18 }
```

## 2. Access Event published state

# HASH PATH FILTERING - EXERCISE (1)

```
1 {  
2   "Event": {  
3     "uuid": ...  
4     "distribution": 1,  
5     "published": false,  
6     "Attribute": [  
7       {  
8         "type": "ip-src",  
9         "value": "8.8.8.8", ...  
10      },  
11      {  
12        "type": "domain",  
13        "value": "misp-project.org", ...  
14      }  
15    ],  
16    ...  
17  }  
18 }
```

## 2. Access Event published state

- ▶ Event.published

## HASH PATH FILTERING - EXERCISE (2)

```
1 {  
2   "Event": {  
3     "uuid": ...  
4     "distribution": 1,  
5     "published": false,  
6     "Attribute": [  
7       {  
8         "type": "ip-src",  
9         "value": "8.8.8.8", ...  
10      },  
11      {  
12        "type": "domain",  
13        "value": "misp-project.org", ...  
14      }  
15    ],  
16    ...  
17  }  
18 }
```

### 3. Access all Attribute types

- ▶ Hint: Use `{n}` to loop

## HASH PATH FILTERING - EXERCISE (2)

```
1 {
2   "Event": {
3     "uuid": ...
4     "distribution": 1,
5     "published": false,
6     "Attribute": [
7       {
8         "type": "ip-src",
9         "value": "8.8.8.8", ...
10      },
11      {
12        "type": "domain",
13        "value": "misp-project.org", ...
14      }
15    ],
16    ...
17  }
18 }
```

### 3. Access all Attribute types

- ▶ Hint: Use `{n}` to loop
- ▶ `Event.Attribute.{n}.type`

## HASH PATH FILTERING - EXERCISE (3)

```
1 {  
2   "Event": {  
3     "Attribute": [  
4       {  
5         "type": "ip-src",  
6         "value": "8.8.8.8",  
7         "Tag": [  
8           {  
9             "name": "PAP:AMBER", ...  
10          }  
11        ], ...  
12      }  
13    ],  
14    ...  
15  }  
16 }
```

### 3. Access all Tags attached to Attributes



## HASH PATH FILTERING - EXERCISE (3)

```
1 {  
2   "Event": {  
3     "Attribute": [  
4       {  
5         "type": "ip-src",  
6         "value": "8.8.8.8",  
7         "Tag": [  
8           {  
9             "name": "PAP:AMBER", ...  
10          }  
11        ], ...  
12      }  
13    ],  
14    ...  
15  }  
16 }
```

### 3. Access all Tags attached to Attributes

- ▶ `Event.Attribute.{n}.Tag.{n}.name`

# HASH PATH FILTERING - EXERCISE (4)

```
1 {
2     "Event": {
3         "Tag": [
4             {
5                 "name": "tlp:green", ...
6             }
7         ], ...
8         "Attribute": [
9             {
10                "value": "8.8.8.8",
11                "Tag": [
12                    {
13                        "name": "PAP:AMBER", ...
14                    }
15                ], ...
16            }
17        ],
18    }
19 }
```

4. Access all Tags attached to Attributes and from the Event

► Hint: Use **`_allTags`** to access **all** tags

# HASH PATH FILTERING - EXERCISE (4)

```
1 {
2   "Event": {
3     "Tag": [
4       {
5         "name": "tlp:green", ...
6       }
7     ], ...
8     "Attribute": [
9       {
10        "value": "8.8.8.8",
11        "Tag": [
12          {
13            "name": "PAP:AMBER", ...
14          }
15        ], ...
16      }
17    ],
18  }
19 }
```

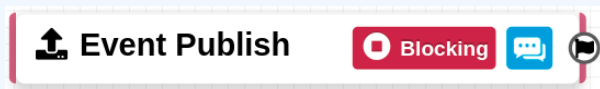
4. Access all Tags attached to Attributes and from the Event
- ▶ `Event.Attribute.{n}._allTags.{n}.name`

# HASH PATH FILTERING - EXERCISE (4)

```
1 {
2   "Event": {
3     "Tag": [...],
4     "Attribute": [
5       {
6         "value": "8.8.8.8",
7         "_allTags": [
8           {
9             "name": "tlp:green",
10            "inherited": true, ...
11          },
12          {
13            "name": "PAP:AMBER",
14            "inherited": false, ...
15          }
16        ],
17      }
18    ]
19  }
```

## 4. Access all Tags attached to Attributes and from the Event

- ▶ `Event.Attribute.{n}._allTags.{n}.name`



- In most cases, the format is the **MISP Core format**
  - ▶ Attributes are **always encapsulated** in the Event or Object
- The MISP Core format has **additional properties**
  - ▶ Additional key **\_AttributeFlattened**
  - ▶ Additional key **\_allTags**
  - ▶ Additional key **inherited** for Tags

# DEBUGGING

# DEBUGGING WORKFLOWS: LOG ENTRIES

- Workflow execution is logged in the application logs:
  - ▶ `/admin/logs/index`
  - ▶ ⚠ Might be phased out as its too verbose
- Or stored on disk in the following file:
  - ▶ `/app/tmp/logs/workflow-execution.log`

## Logs

« previous

next »

Emails

Authentication issues


MISP Update results

Setting changes

Warnings and errors

| Id ↑  | Email  | Org    | Created             | Model    | Model ID | Action           | Title   |
|-------|--------|--------|---------------------|----------|----------|------------------|---|
| 49146 | SYSTEM | SYSTEM | 2022-08-01 07:34:40 | Workflow | 162      | execute_workflow | Finished executing workflow for trigger `enrichment-before-query` (162). Outcome: success |
| 49144 | SYSTEM | SYSTEM | 2022-08-01 07:34:39 | Workflow | 162      | execute_workflow | Started executing workflow for trigger `enrichment-before-query` (162)                    |

# DEBUGGING WORKFLOWS: DEBUG MODE

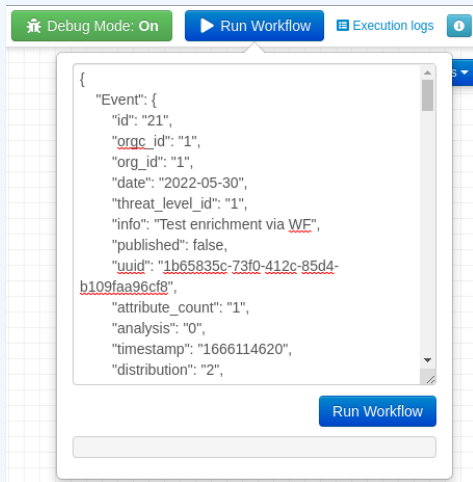
- The  can be turned on for each workflows
- Each nodes will send data to the provided URL
  - ▶ Configure the setting: `Plugin.Workflow_debug_url`
- Result can be visualized in
  - ▶ **offline:** `tools/misp-workflows/webhook-listener.py`
  - ▶ **online:** `requestbin.com` or similar websites

| LIVE PAUSE 🔍 Type to search... |      |                                     |
|--------------------------------|------|-------------------------------------|
| Today                          |      |                                     |
| 2:25:10 pm                     | POST | /end?outcome=blocked                |
| 2:25:09 pm                     | POST | /exec/stop-execution?result=success |
| 2:25:09 pm                     | POST | /exec/tag-if?result=success         |
| 2:25:08 pm                     | POST | /init?type=blocking                 |



# DEBUGGING MODULES: RE-RUNNING WORKFLOWS

- Try workflows with custom input
- Re-run workflows to ease debugging



- Workflow **execution and outcome**
- Individual module **execution and outcome**
- **Live** workflow debugging with module inspection
- **Re-running/testing** workflows with custom data

# 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

- More action modules 🎬
- More logic modules ➡
- More triggers 🚩
- Recursion prevention system
- Improvement for logging

- Designed to **quickly** and **cheaply** integrate MISP in CTI pipelines
- Waiting for feedback!
  - ▶ New triggers?
  - ▶ New modules?