Automation in MISP

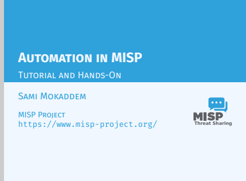
Tutorial and Hands-On

Sami Mokaddem

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

2024-04-15

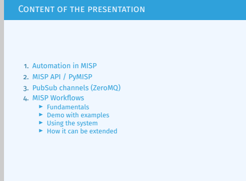
Automation in MISP



Automation in MISP
Tutorial and Hands-On
Sami Mokaddem
MISP Project
https://www.misp-project.org/

Content of the presentation

1. Automation in MISP
2. MISP API / PyMISP
3. PubSub channels (ZeroMQ)
4. MISP Workflows
   - Fundamentals
   - Demo with examples
   - Using the system
   - How it can be extended



Content of the presentation

1. Automation in MISP
2. MISP API / PyMISP
3. PubSub channels (ZeroMQ)
4. 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

Automation in MISP: What already exists?

MISP API / PyMISP

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

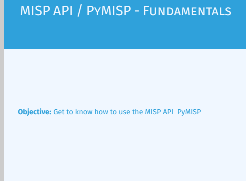
PubSub channels

- After the actions happen: No feedback to MISP
- Tougher to put in place & to share
- Full integration amounts to develop a new tool

MISP API / PyMISP - Fundamentals

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

└MISP API / PyMISP - Fundamentals



MISP API / PyMISP - Fundamentals

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

MISP API / PyMISP - Demo

- Generate an API key
- RestClient overview
- MISP API Overview notebook[1]
- PyMISP Overview notebook[2]

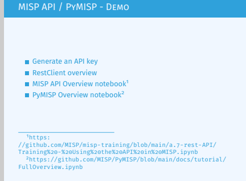
[1] https://github.com/MISP/misp-training/blob/main/a.7-rest-API/Training%20-%20Using%20the%20API%20in%20MISP.ipynb

[2] https://github.com/MISP/PyMISP/blob/main/docs/tutorial/FullOverview.ipynb

2024-04-15

Automation in MISP

└─MISP API / PyMISP - Demo



MISP API / PyMISP - Demo

- Generate an API key
- RestClient overview
- MISP API Overview notebook[1]
- PyMISP Overview notebook[2]

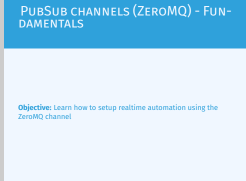
[1] https://github.com/MISP/misp-training/blob/main/a.7-rest-API/Training%20-%20Using%20the%20API%20in%20MISP.ipynb

[2] https://github.com/MISP/PyMISP/blob/main/docs/tutorial/FullOverview.ipynb

PubSub channels (ZeroMQ) - Fundamentals

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

└PubSub channels (ZeroMQ)



PubSub channels (ZeroMQ) - Fundamentals

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

ZeroMQ channel - Demo

- What is ZeroMQ?
  - *N-to-N Asynchronous message-processing tasks*
  - *Publisher (MISP) and consumer (scripts)*
- Configuring ZeroMQ in MISP
- Integrating with the ZeroMQ of MISP

ZeroMQ channel - Demo

- What is ZeroMQ?
  - *N-to-N Asynchronous message-processing tasks*
  - *Publisher (MISP) and consumer (scripts)*
- Configuring ZeroMQ in MISP
- Integrating with the ZeroMQ of MISP

MISP Workflows - Fundamentals

Objective: Learn how to use the MISP Worklfow feature

└MISP Workflows - Fundamentals

MISP Workflows - Fundamentals

Objective: Learn how to use the MISP Worklfow feature

Automation in MISP: What already exists?

MISP API / PyMISP

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

PubSub channels

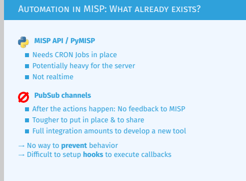
- After the actions happen: No feedback to MISP
- Tougher to put in place & to share
- Full integration amounts to develop a new tool

→ No way to **prevent** behavior
→ Difficult to setup **hooks** to execute callbacks

2024-04-15

Automation in MISP

└─Automation in MISP: What already exists?



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

What type of use-cases are we trying to support?

- **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
  - ...

What type of use-cases are we trying to support?

- **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
  - ...

Simple automation in MISP made easy

- Why?
  - Everyone loves **simple automation**
  - **Visual** dataflow programming
  - Users want **more control**
- How?
  - **Drag & Drop** editor
  - Prevent actions **before they happen**
  - Flexible **Plug & Play** system
  - **Share** workflows, **debug** and **replay**

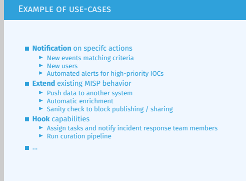
└─Simple automation in MISP made easy

Simple automation in MISP made easy

- Why?
  - Everyone loves **simple automation**
  - **Visual** dataflow programming
  - Users want **more control**
- How?
  - **Drag & Drop** editor
  - Prevent actions **before they happen**
  - 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
- **Hook** capabilities
  - Assign tasks and notify incident response team members
  - Run curation pipeline
- ...



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
- **Hook** capabilities
  - Assign tasks and notify incident response team members
  - Run curation pipeline
- ...

Workflow - Fundamentals

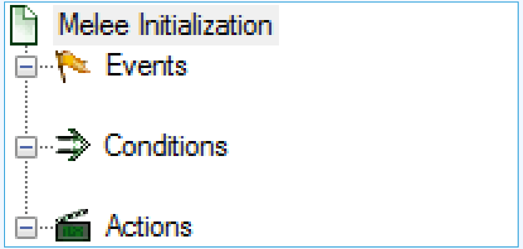
Objective: Start with the foundation to understand the basics

└Workflow - Fundamentals

Workflow - Fundamentals

Objective: Start with the foundation to understand the basics

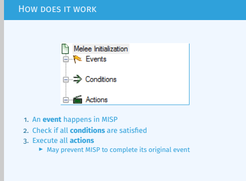
How does it work



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

Automation in MISP

└How does it work


How does it work

Melee Initialization
Events
Conditions
Actions

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

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**

What kind of events?

Events

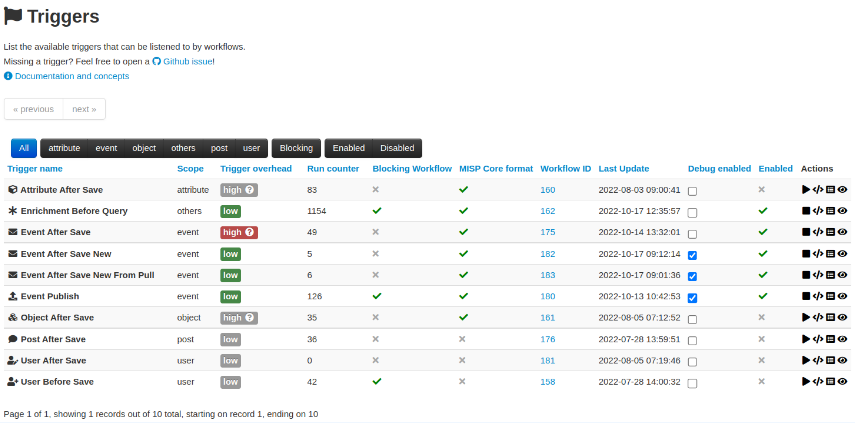
- New MISP Event
- Attribute has been saved
- New discussion post
- New user created
- Query against third-party services
- ...

? Supported events in MISP are called **Triggers**

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

Triggers currently available

Currently 10 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 »

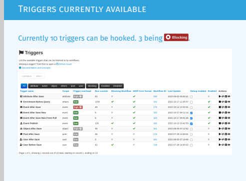
Trigger name	Scope	Trigger overhead	Run counter	Blocking Workflow	MISP Core format	Workflow ID	Last Update	Debug enabled	Enabled	Actions
Attribute After Save	attribute	high ?	83	✗	✓	160	2022-08-03 09:00:41	□	✗	▶ </> ☰ 👁
Enrichment Before Query	others	low	1154	✓	✓	162	2022-10-17 12:35:57	□	✓	■ </> ☰ 👁
Event After Save	event	high ?	49	✗	✓	175	2022-10-14 13:32:01	□	✓	■ </> ☰ 👁
Event After Save New	event	low	5	✗	✓	182	2022-10-17 09:12:14	☑	✓	■ </> ☰ 👁
Event After Save New From Pull	event	low	6	✗	✓	183	2022-10-17 09:01:36	☑	✓	■ </> ☰ 👁
Event Publish	event	low	126	✓	✓	180	2022-10-13 10:42:53	☑	✓	■ </> ☰ 👁
Object After Save	object	high ?	35	✗	✓	161	2022-08-05 07:12:52	□	✗	▶ </> ☰ 👁
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	2022-07-28 14:00:32	□	✗	▶ </> ☰ 👁

Page 1 of 1, showing 1 records out of 10 total, starting on record 1, ending on 10

2024-04-15

Automation in MISP

└─Triggers currently available

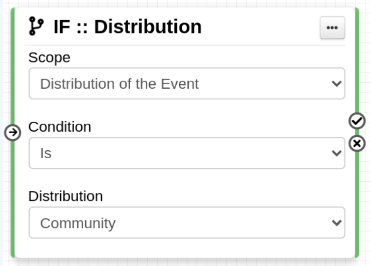


What kind of conditions?

⇛ 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

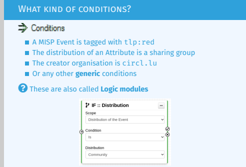
? These are also called **Logic modules**


IF :: Distribution

Scope: Distribution of the Event

Condition: Is

Distribution: Community


What kind of conditions?

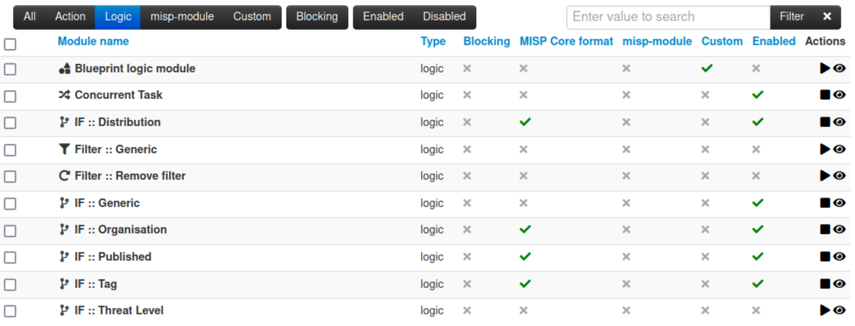
⇛ Conditions

- 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

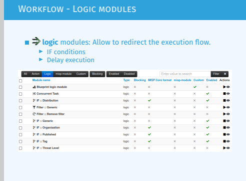


□	Module name	Type	Blocking	MISP Core format	misp-module	Custom	Enabled	Actions
□	Blueprint logic module	logic	×	×	×	✓	×	▶ 👁
□	Concurrent Task	logic	×	×	×	×	✓	■ 👁
□	IF :: Distribution	logic	×	✓	×	×	✓	■ 👁
□	Filter :: Generic	logic	×	×	×	×	×	▶ 👁
□	Filter :: Remove filter	logic	×	×	×	×	×	▶ 👁
□	IF :: Generic	logic	×	×	×	×	✓	■ 👁
□	IF :: Organisation	logic	×	✓	×	×	✓	■ 👁
□	IF :: Published	logic	×	✓	×	×	✓	■ 👁
□	IF :: Tag	logic	×	✓	×	×	✓	■ 👁
□	IF :: Threat Level	logic	×	×	×	×	×	▶ 👁

2024-04-15

Automation in MISP

└─Workflow - Logic modules

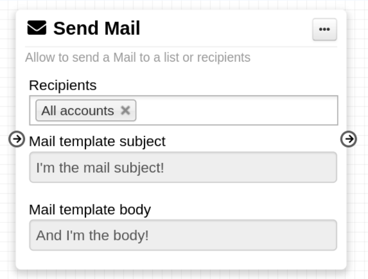


What kind of actions?

Actions

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

These are also called **Action modules**



Send Mail

Allow to send a Mail to a list or recipients

Recipients: All accounts ×

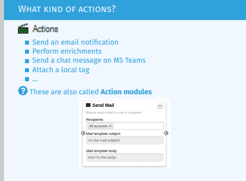
Mail template subject: I'm the mail subject!

Mail template body: And I'm the body!

2024-04-15

Automation in MISP

└What kind of actions?



What kind of actions?

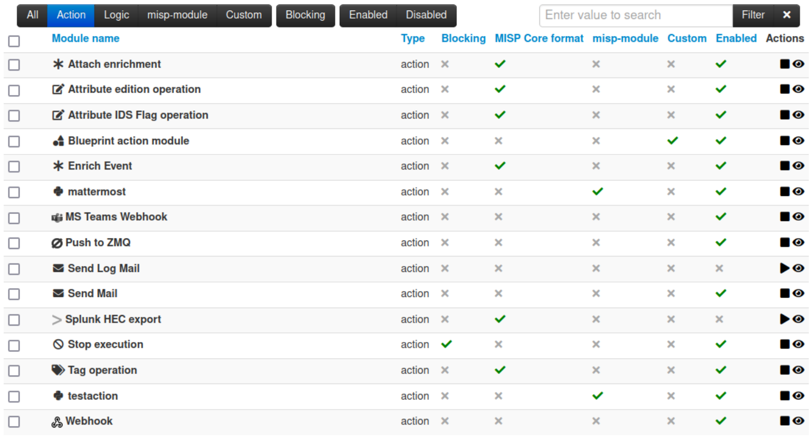
Actions

- 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

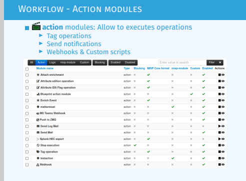


□	Module name	Type	Blocking	MISP Core format	misp-module	Custom	Enabled	Actions
□	Attach enrichment	action	✕	✓	✕	✕	✓	■ 👁
□	Attribute edition operation	action	✕	✓	✕	✕	✓	■ 👁
□	Attribute IDS Flag operation	action	✕	✓	✕	✕	✓	■ 👁
□	Blueprint action module	action	✕	✕	✕	✓	✓	■ 👁
□	Enrich Event	action	✕	✓	✕	✕	✓	■ 👁
□	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	✕	✕	✕	✕	✓	■ 👁

2024-04-15

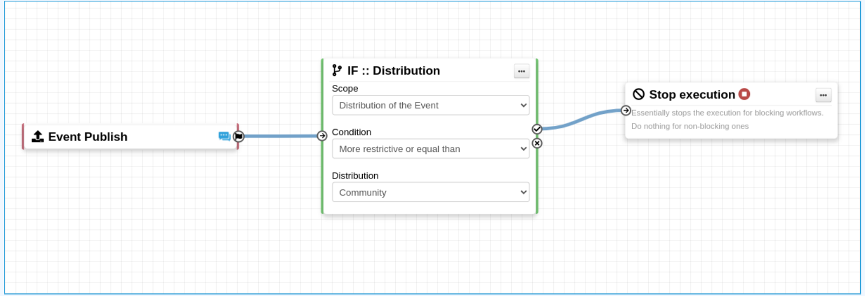
Automation in MISP

└Workflow - Action modules



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**



Event Publish

IF :: Distribution

Scope: Distribution of the Event

Condition: More restrictive or equal than

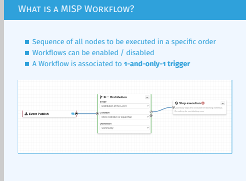
Distribution: Community

Stop execution

Essentially stops the execution for blocking workflows.
Do nothing for non-blocking ones

2024-04-15

└What is a MISP Workflow?

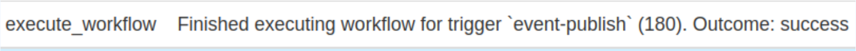


What is a MISP Workflow?

- 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**

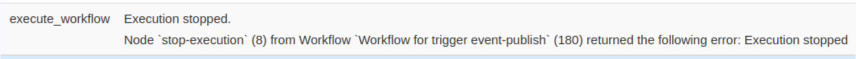
Workflow execution for Event publish

- 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
```

└Workflow execution for Event publish

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
failure | blocked: Stop publishing and log the reason

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

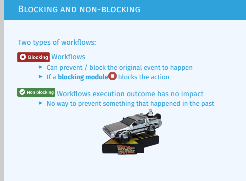
✓ Non blocking Workflows execution outcome has no impact

- No way to prevent something that happened in the past

2024-04-15

Automation in MISP

└─Blocking and non-blocking


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

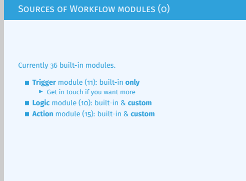
✓ Non blocking Workflows execution outcome has no impact

- No way to prevent something that happened in the past

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 (15): built-in & **custom**



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 (15): built-in & **custom**

Sources of Workflow modules (1)

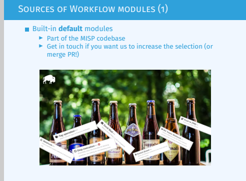
- Built-in **default** modules
  - Part of the MISP codebase
  - Get in touch if you want us to increase the selection (or merge PR!)

Tag operation v0.2
Add or remove tags on Event or Attributes.
Webhook v0.2
Allow to perform custom callbacks to the provided URL
Stop execution v0.1
Essentially stops the execution for blocking workflows. Do nothing for non-blocking ones
Send Mail v0.1
Allow to send a Mail to a list of recipients
Enrich Event v0.1
Enrich all Attributes contained in the Event with the provided module
MS Teams Webhook v0.2
Perform callbacks to the MS Teams webhook provided by the "Incoming Webhook" connector
Push to ZMQ v0.1
Push to the ZMQ channel

2024-04-15

Automation in MISP

└Sources of Workflow modules (1)

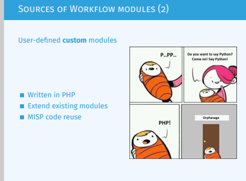

Sources of Workflow modules (1)

- 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

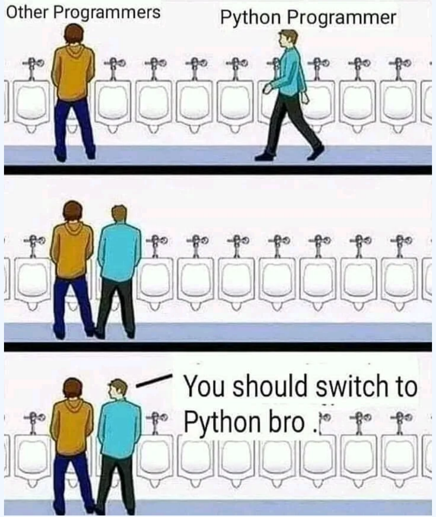
- Written in PHP
- Extend existing modules
- MISP code reuse

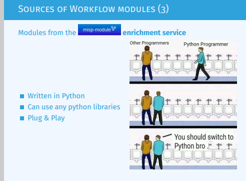

Sources of Workflow modules (2)
User-defined **custom** modules
- Written in PHP
- Extend existing modules
- MISP code reuse

Sources of Workflow modules (3)

Modules from the misp-module enrichment service

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

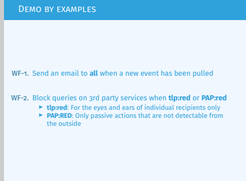



Sources of Workflow modules (3)
Modules from the misp-module enrichment service
- Written in Python
- Can use any python libraries
- Plug & Play

Demo by examples

WF-1. Send an email to **all** 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 by examples

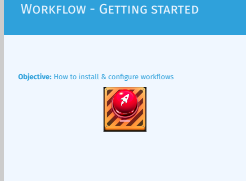
WF-1. Send an email to **all** when a new event has been pulled

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

Workflow - Getting started

Objective: How to install & configure workflows

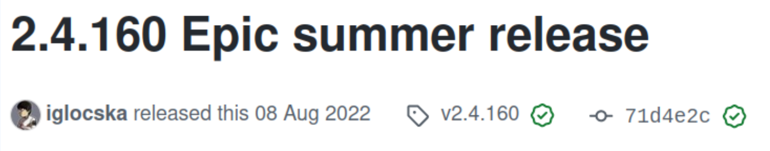
└Workflow - Getting started



Workflow - Getting started

Objective: How to install & configure workflows

Getting started with workflows (1)



2.4.160 Epic summer release

iglocska released this 08 Aug 2022 v2.4.160 ✓ 71d4e2c ✓

1. Update your MISP server
2. Update all your sub-modules

Getting started with workflows (1)

2.4.160 Epic summer release

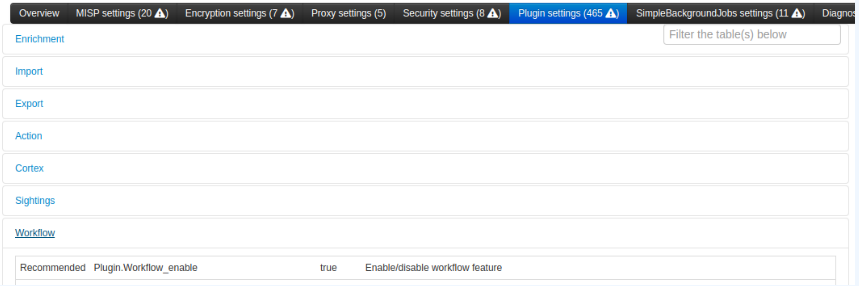
iglocska released this 08 Aug 2022 v2.4.160 71d4e2c

1. Update your MISP server
2. Update all your sub-modules

Getting started with workflows (2)

Review MISP settings:

1. Make sure `MISP.background_jobs` is turned on
2. Make sure workers are up-and-running and healthy
3. Turn the setting `Plugin.Workflow_enable` on



Enrichment

Import

Export

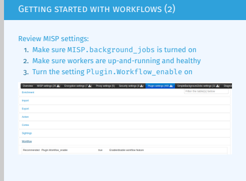
Action

Cortex

Sightings

Workflow

Recommended	Plugin.Workflow_enable	true	Enable/disable workflow feature



Getting started with workflows (2)

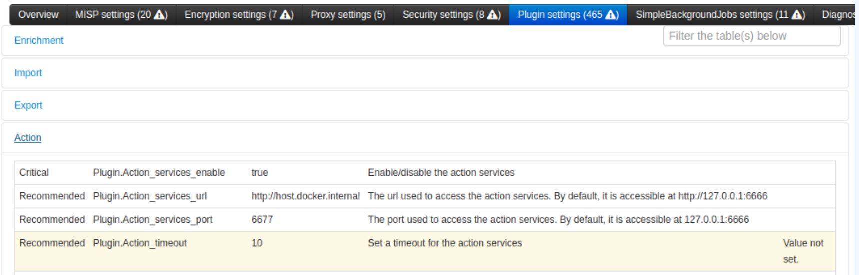
Review MISP settings:

1. Make sure `MISP.background_jobs` is turned on
2. Make sure workers are up-and-running and healthy
3. Turn the setting `Plugin.Workflow_enable` on

GETTING STARTED WITH WORKFLOWS (3)

Review MISP settings:

4. [optional:misp-module] Turn the setting `Plugin.Action_services_enable` on

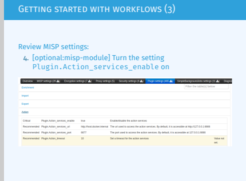


Critical	Plugin.Action_services_enable	true	Enable/disable the action services	
Recommended	Plugin.Action_services_url	http://host.docker.internal	The url used to access the action services. By default, it is accessible at http://127.0.0.1:6666	
Recommended	Plugin.Action_services_port	6677	The port used to access the action services. By default, it is accessible at 127.0.0.1:6666	
Recommended	Plugin.Action_timeout	10	Set a timeout for the action services	Value not set.

2024-04-15

Automation in MISP

└Getting started with workflows (3)


GETTING STARTED WITH WORKFLOWS (3)

Review MISP settings:

4. [optional:misp-module] Turn the setting `Plugin.Action_services_enable` on

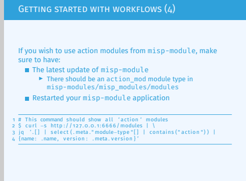
GETTING STARTED WITH WORKFLOWS (4)

If you wish to use action modules from `misp-module`, make sure to have:

- The latest update of `misp-module`
  - There should be an `action_mod` module type in `misp-modules/misp_modules/modules`
- Restarted your `misp-module` application

```
# This command should show all 'action' modules
$ curl -s http://127.0.0.1:6666/modules | \
jq '.[] | select(.meta."module-type"[] | contains("action")) |
{name: .name, version: .meta.version}'
```

└─Getting started with workflows (4)



GETTING STARTED WITH WORKFLOWS (4)

If you wish to use action modules from misp-module, make sure to have:

- The latest update of misp-module
  - There should be an action_mod module type in misp-modules/misp_modules/modules
- Restarted your misp-module application

# This command should show all 'action' modules
$ curl -s http://127.0.0.1:6666/modules | \
jq '.[] | select(.meta."module-type"[] | contains("action")) |
{name: .name, version: .meta.version}'

Getting started with workflows (5)

Everything is ready?

Let’s see how to build a workflow!

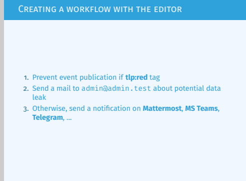
Getting started with workflows (5)

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
2. Send a mail to `admin@admin.test` about potential data leak
3. Otherwise, send a notification on **Mattermost**, **MS Teams**, **Telegram**, ...



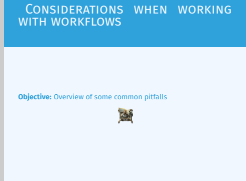
Creating a workflow with the editor

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

CONSIDERATIONS WHEN WORKING WITH WORKFLOWS

Objective: Overview of some common pitfalls

└Considerations when

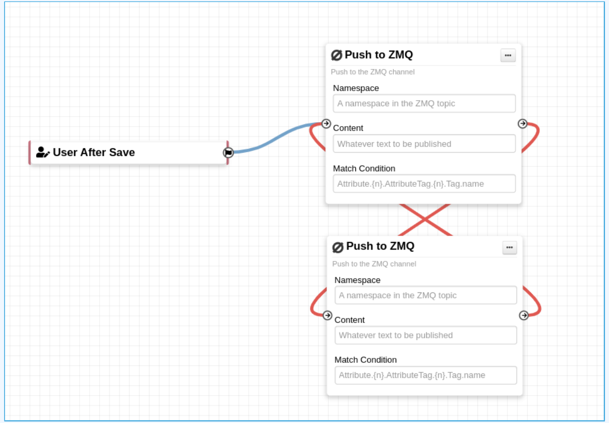


CONSIDERATIONS WHEN WORKING WITH WORKFLOWS

Objective: Overview of some common pitfalls

Working with the editor - Operations not allowed

Execution loop are not authorized



User After Save

Push to ZMQ

Push to the ZMQ channel

Namespace: A namespace in the ZMQ topic

Content: Whatever text to be published

Match Condition: Attribute.{n}.AttributeTag.{n}.Tag.name

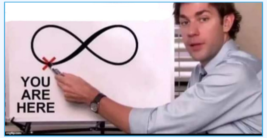
Push to ZMQ

Push to the ZMQ channel

Namespace: A namespace in the ZMQ topic

Content: Whatever text to be published

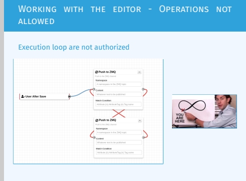
Match Condition: Attribute.{n}.AttributeTag.{n}.Tag.name



2024-04-15

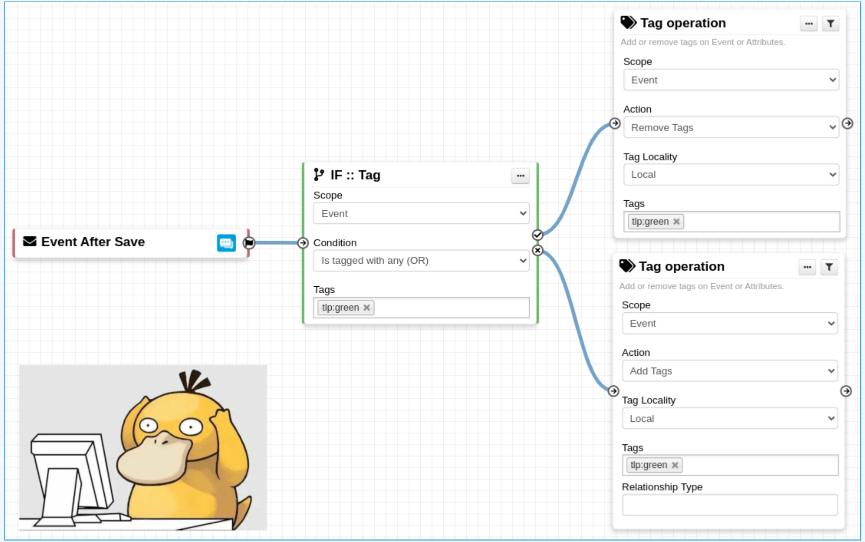
Automation in MISP

└─Working with the editor - Operations not allowed


Working with the editor - Operations not allowed

Execution loop are not authorized

Recursive workflows



Event After Save

IF :: Tag

Scope: Event

Condition: Is tagged with any (OR)

Tags: tlp:green ×

Tag operation

Add or remove tags on Event or Attributes.

Scope: Event

Action: Remove Tags

Tag Locality: Local

Tags: tlp:green ×

Tag operation

Add or remove tags on Event or Attributes.

Scope: Event

Action: Add Tags

Tag Locality: Local

Tags: tlp:green ×

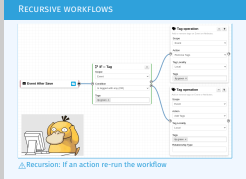
Relationship Type:

⚠Recursion: If an action re-run the workflow

2024-04-15

Automation in MISP

└Recursive workflows

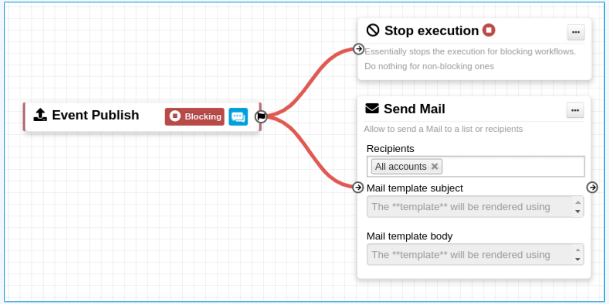


Recursive workflows

⚠Recursion: If an action re-run the workflow

Working with the editor - Operations not allowed

Multiple connections from the same output



Stop execution

Essentially stops the execution for blocking workflows. Do nothing for non-blocking ones

Event Publish

Blocking

Send Mail

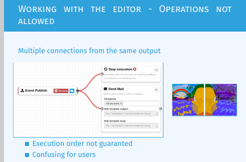
Allow to send a Mail to a list or recipients

Recipients: All accounts ×

Mail template subject: The **template** will be rendered using

Mail template body: The **template** will be rendered using

- Execution order not guaranted
- Confusing for users



Working with the editor - Operations not allowed

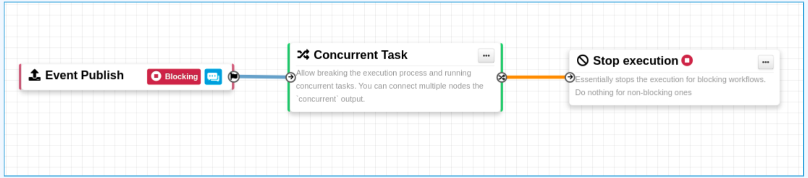
Multiple connections from the same output

- Execution order not guaranted
- Confusing for users

Working with the editor

Cases showing a warning:

- **Blocking** modules ⏹ in a ✓ Non blocking workflow
- **Blocking** modules ⏹ after a **concurrent tasks** module



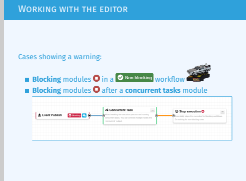
Event Publish ⏹ Blocking

Concurrent Task

Allow breaking the execution process and running concurrent tasks. You can connect multiple nodes the `concurrent` output.

Stop execution ⏹

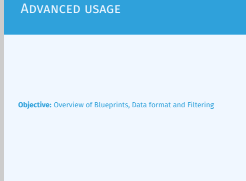
Essentially stops the execution for blocking workflows. Do nothing for non-blocking ones



Advanced usage

Objective: Overview of Blueprints, Data format and Filtering

└Advanced usage

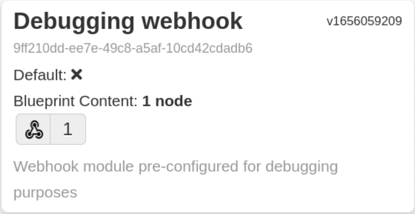


Advanced usage

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: ✗

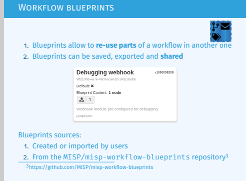
Blueprint Content: **1 node**

Webhook module pre-configured for debugging purposes

Blueprints sources:

1. Created or imported by users
2. From the `MISP/misp-workflow-blueprints` repository[3]

[3] https://github.com/MISP/misp-workflow-blueprints



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:

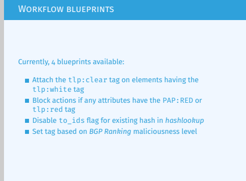
1. Created or imported by users
2. From the MISP/misp-workflow-blueprints repository[3]

[3] https://github.com/MISP/misp-workflow-blueprints

Workflow blueprints

Currently, 4 blueprints available:

- Attach the `tlp:clear` tag on elements having the `tlp:white` tag
- Block actions if any attributes have the `PAP:RED` or `tlp:red` tag
- Disable `to_ids` flag for existing hash in *hashlookup*
- Set tag based on *BGP Ranking* maliciousness level



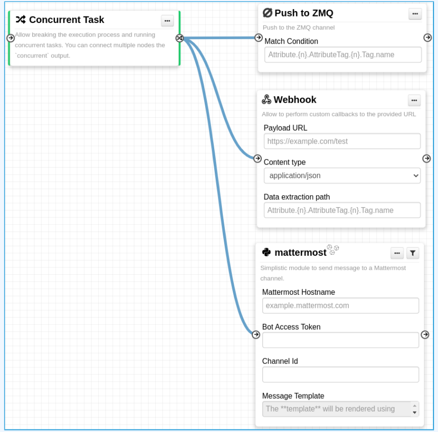
Workflow blueprints

Currently, 4 blueprints available:

- Attach the `tlp:clear` tag on elements having the `tlp:white` tag
- Block actions if any attributes have the `PAP:RED` or `tlp:red` tag
- Disable `to_ids` flag for existing hash in *hashlookup*
- Set tag based on *BGP Ranking* maliciousness level

Logic module: Concurrent Task

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

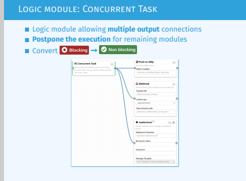


Concurrent Task
Allow breaking the execution process and running concurrent tasks. You can connect multiple nodes the `concurrent` output.

Push to ZMQ
Push to the ZMQ channel
Match Condition
Attribute.{n}.AttributeTag.{n}.Tag.name

Webhook
Allow to perform custom callbacks to the provided URL
Payload URL
https://example.com/test
Content type
application/json
Data extraction path
Attribute.{n}.AttributeTag.{n}.Tag.name

mattermost
Simplistic module to send message to a Mattermost channel.
Mattermost Hostname
example.mattermost.com
Bot Access Token
Channel Id
Message Template
The **template** will be rendered using



Data format in Workflows

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

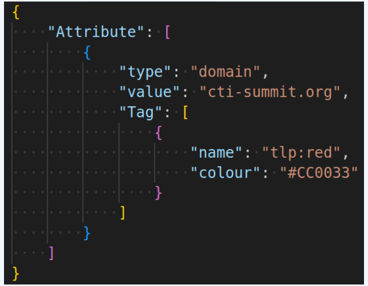
Data format in Workflows

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

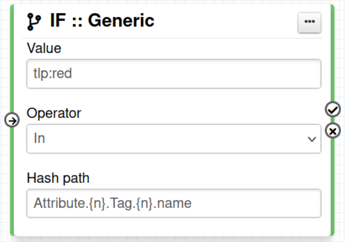
Hash path filtering (1)

Filtering and checking conditions using hash path expression.

```
$path_expression = '{n}[name=fred].id';
$users = [
    {'id': 123, 'name': 'fred', 'surname': 'bloggs'},
    {'id': 245, 'name': 'fred', 'surname': 'smith'},
    {'id': 356, 'name': 'joe', 'surname': 'smith'},
];
$ids = Hash::extract($users, $path_expression);
// => $ids will be [123, 245]
```



```
{
    "Attribute": [
        {
            "type": "domain",
            "value": "cti-summit.org",
            "Tag": [
                {
                    "name": "tlp:red",
                    "colour": "#CC0033"
                }
            ]
        }
    ]
}
```



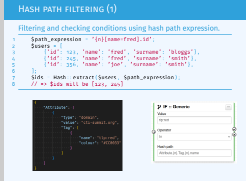
IF :: Generic

Value: tlp:red

Operator: In

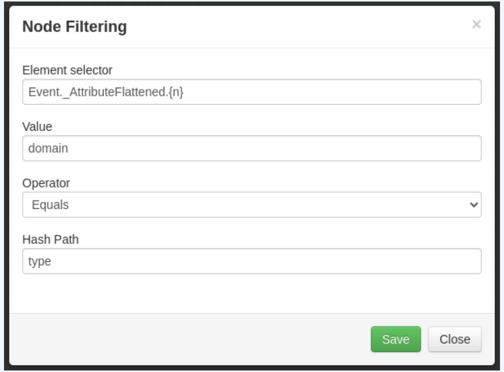
Hash path: Attribute.{n}.Tag.{n}.name

└Hash path filtering (1)



Hash path filtering (2)

Hash path filtering can be used to **filter** data **on the node** it is passed to or on the **execution path**.



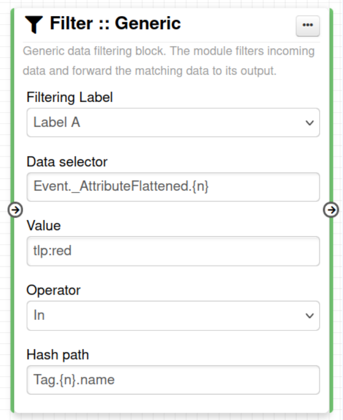
Node Filtering

Element selector: Event._AttributeFlattened.{n}

Value: domain

Operator: Equals

Hash Path: type



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

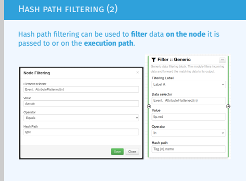
Operator: In

Hash path: Tag.{n}.name

2024-04-15

Automation in MISP

└Hash path filtering (2)



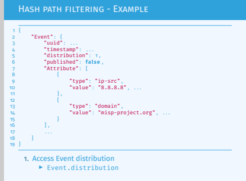
Hash path filtering - Example

```
{
    "Event": {
        "uuid": ...
        "timestamp": ...
        "distribution": 1,
        "published": false ,
        "Attribute": [
            {
                "type": "ip-src",
                "value": "8.8.8.8", ...
            },
            {
                "type": "domain",
                "value": "misp-project.org", ...
            }
        ],
        ...
    }
}
```

1. Access Event distribution
    - `Event.distribution`

2024-04-15

└─Hash path filtering - Example



Hash path filtering - Example

```
{
    "Event": {
        "uuid": ...
        "timestamp": ...
        "distribution": 1,
        "published": false ,
        "Attribute": [
            {
                "type": "ip-src",
                "value": "8.8.8.8", ...
            },
            {
                "type": "domain",
                "value": "misp-project.org", ...
            }
        ],
        ...
    }
}
```

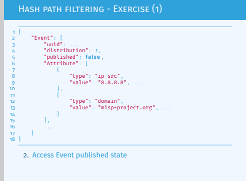
1. Access Event distribution
    - Event.distribution

Hash path filtering - Exercise (1)

```
{
    "Event": {
        "uuid": ...
        "distribution": 1,
        "published": false,
        "Attribute": [
            {
                "type": "ip-src",
                "value": "8.8.8.8", ...
            },
            {
                "type": "domain",
                "value": "misp-project.org", ...
            }
        ],
        ...
    }
}
```

2. Access Event published state

└─Hash path filtering - Exercise (1)



Hash path filtering - Exercise (1)

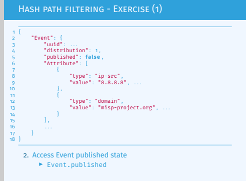
```
{
    "Event": {
        "uuid": ...
        "distribution": 1,
        "published": false,
        "Attribute": [
            {
                "type": "ip-src",
                "value": "8.8.8.8", ...
            },
            {
                "type": "domain",
                "value": "misp-project.org", ...
            }
        ],
        ...
    }
}
```

2. Access Event published state

Hash path filtering - Exercise (1)

```
{
    "Event": {
        "uuid": ...
        "distribution": 1,
        "published": false,
        "Attribute": [
            {
                "type": "ip-src",
                "value": "8.8.8.8", ...
            },
            {
                "type": "domain",
                "value": "misp-project.org", ...
            }
        ],
        ...
    }
}
```

2. Access Event published state
   - `Event.published`



Hash path filtering - Exercise (1)

```
{
    "Event": {
        "uuid": ...
        "distribution": 1,
        "published": false,
        "Attribute": [
            {
                "type": "ip-src",
                "value": "8.8.8.8", ...
            },
            {
                "type": "domain",
                "value": "misp-project.org", ...
            }
        ],
        ...
    }
}
```

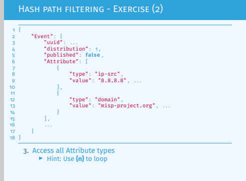
2. Access Event published state
   - Event.published

Hash path filtering - Exercise (2)

```
{
    "Event": {
        "uuid": ...
        "distribution": 1,
        "published": false,
        "Attribute": [
            {
                "type": "ip-src",
                "value": "8.8.8.8", ...
            },
            {
                "type": "domain",
                "value": "misp-project.org", ...
            }
        ],
        ...
    }
}
```

3. Access all Attribute types
    - Hint: Use **{n}** to loop

└─Hash path filtering - Exercise (2)



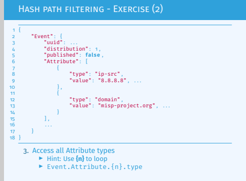
Hash path filtering - Exercise (2)

3. Access all Attribute types
    - Hint: Use {n} to loop

Hash path filtering - Exercise (2)

```
{
    "Event": {
        "uuid": ...
        "distribution": 1,
        "published": false,
        "Attribute": [
            {
                "type": "ip-src",
                "value": "8.8.8.8", ...
            },
            {
                "type": "domain",
                "value": "misp-project.org", ...
            }
        ],
        ...
    }
}
```

3. Access all Attribute types
   - Hint: Use **{n}** to loop
   - `Event.Attribute.{n}.type`

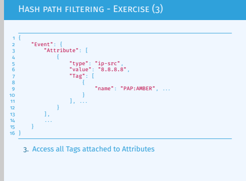

Hash path filtering - Exercise (2)

3. Access all Attribute types
   - Hint: Use {n} to loop
   - Event.Attribute.{n}.type

Hash path filtering - Exercise (3)

```
{
    "Event": {
        "Attribute": [
            {
                "type": "ip-src",
                "value": "8.8.8.8",
                "Tag": [
                    {
                        "name": "PAP:AMBER", ...
                    }
                ], ...
            }
        ],
        ...
    }
}
```

3. Access all Tags attached to Attributes



Hash path filtering - Exercise (3)

```
{
    "Event": {
        "Attribute": [
            {
                "type": "ip-src",
                "value": "8.8.8.8",
                "Tag": [
                    {
                        "name": "PAP:AMBER", ...
                    }
                ], ...
            }
        ],
        ...
    }
}
```

3. Access all Tags attached to Attributes

Hash path filtering - Exercise (3)

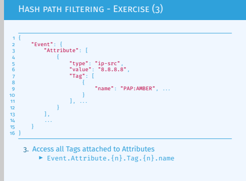
```
{
    "Event": {
        "Attribute": [
            {
                "type": "ip-src",
                "value": "8.8.8.8",
                "Tag": [
                    {
                        "name": "PAP:AMBER", ...
                    }
                ], ...
            }
        ],
        ...
    }
}
```

3. Access all Tags attached to Attributes
   - ▶ `Event.Attribute.{n}.Tag.{n}.name`

2024-04-15

Automation in MISP

└Hash path filtering - Exercise (3)



Hash path filtering - Exercise (3)

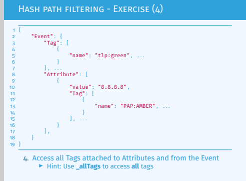
```
{
    "Event": {
        "Attribute": [
            {
                "type": "ip-src",
                "value": "8.8.8.8",
                "Tag": [
                    {
                        "name": "PAP:AMBER", ...
                    }
                ], ...
            }
        ],
        ...
    }
}
```

3. Access all Tags attached to Attributes
   - ▶ Event.Attribute.{n}.Tag.{n}.name

Hash path filtering - Exercise (4)

```
{
    "Event": {
        "Tag": [
            {
                "name": "tlp:green", ...
            }
        ], ...
        "Attribute": [
            {
                "value": "8.8.8.8",
                "Tag": [
                    {
                        "name": "PAP:AMBER", ...
                    }
                ], ...
            }
        ],
    }
}
```

4. Access all Tags attached to Attributes and from the Event
   - Hint: Use **_allTags** to access **all** tags



Hash path filtering - Exercise (4)

4. Access all Tags attached to Attributes and from the Event
   - Hint: Use **_allTags** to access **all** tags

Hash path filtering - Exercise (4)

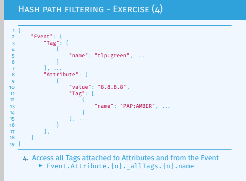
```
{
    "Event": {
        "Tag": [
            {
                "name": "tlp:green", ...
            }
        ], ...
        "Attribute": [
            {
                "value": "8.8.8.8",
                "Tag": [
                    {
                        "name": "PAP:AMBER", ...
                    }
                ], ...
            }
        ],
    }
}
```

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

Automation in MISP

2024-04-15

└ Hash path filtering - Exercise (4)



Hash path filtering - Exercise (4)

```
{
    "Event": {
        "Tag": [
            {
                "name": "tlp:green", ...
            }
        ], ...
        "Attribute": [
            {
                "value": "8.8.8.8",
                "Tag": [
                    {
                        "name": "PAP:AMBER", ...
                    }
                ], ...
            }
        ],
    }
}
```

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

Hash path filtering - Exercise (4)

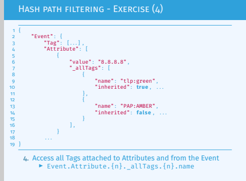
```
{
    "Event": {
        "Tag": [...],
        "Attribute": [
            {
                "value": "8.8.8.8",
                "_allTags": [
                    {
                        "name": "tlp:green",
                        "inherited": true, ...
                    },
                    {
                        "name": "PAP:AMBER",
                        "inherited": false, ...
                    }
                ],
            }
        ...
}
```

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

Automation in MISP

2024-04-15

└Hash path filtering - Exercise (4)



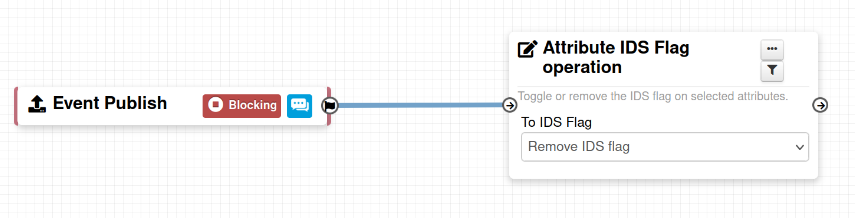
Hash path filtering - Exercise (4)

```
{
    "Event": {
        "Tag": [...],
        "Attribute": [
            {
                "value": "8.8.8.8",
                "_allTags": [
                    {
                        "name": "tlp:green",
                        "inherited": true, ...
                    },
                    {
                        "name": "PAP:AMBER",
                        "inherited": false, ...
                    }
                ],
            }
        ...
}
```

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

Fitlering data on which to apply a module

What happens when an Event is about to be published?



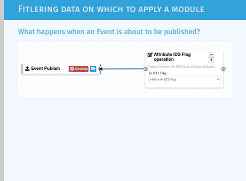
Toggle or remove the IDS flag on selected attributes.

To IDS Flag: Remove IDS flag

Automation in MISP

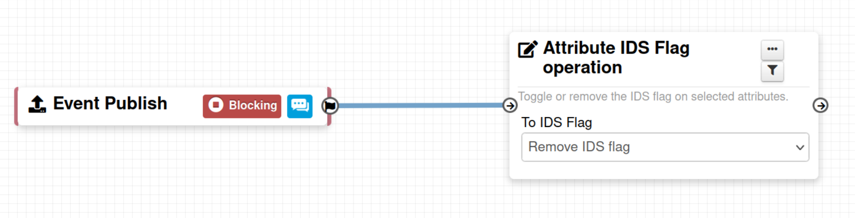
2024-04-15

└Fitlering data on which to apply a module



Fitlering data on which to apply a module

What happens when an Event is about to be published?



Toggle or remove the IDS flag on selected attributes.

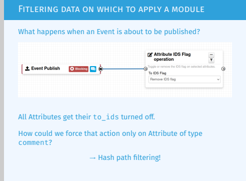
To IDS Flag: Remove IDS flag

All Attributes get their `to_ids` turned off.

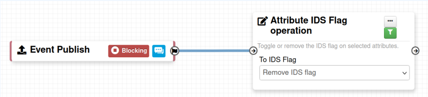
How could we force that action only on Attribute of type `comment`?

→ Hash path filtering!

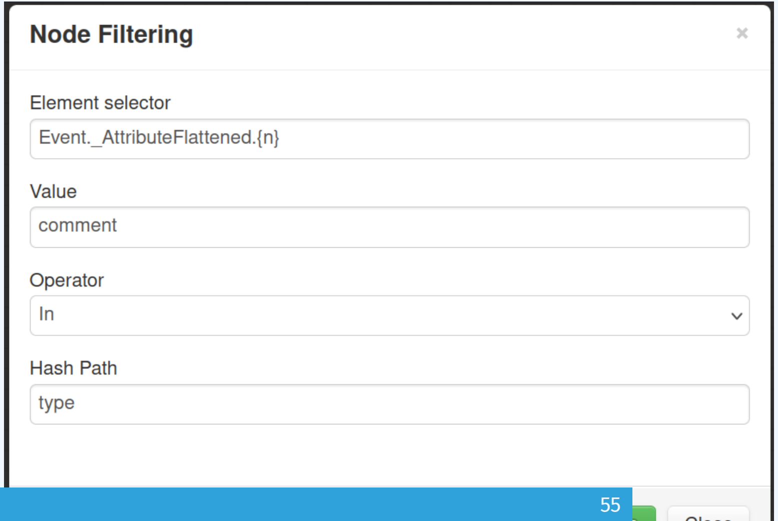
└ Fitlering data on which to apply a module


Fitlering data on which to apply a module
What happens when an Event is about to be published?
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


Event Publish
Blocking
Attribute IDS Flag operation
Toggle or remove the IDS flag on selected attributes.

To IDS Flag: Remove IDS flag



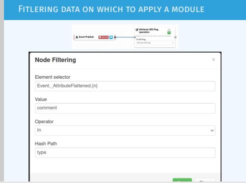
Node Filtering

Element selector: Event._AttributeFlattened.{n}

Value: comment

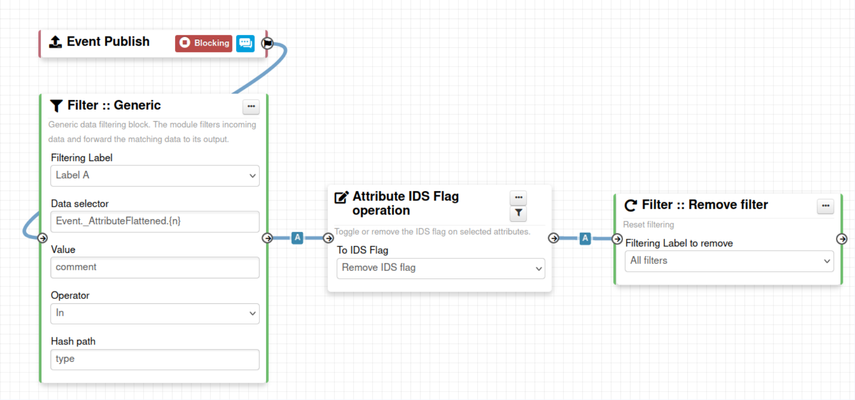
Operator: In

Hash Path: type



Fitlering data on which to apply on multiple modules

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



Event Publish — Blocking

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: comment

Operator: In

Hash path: type

Attribute IDS Flag operation

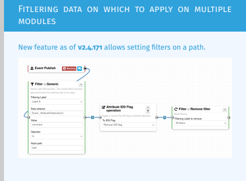
Toggle or remove the IDS flag on selected attributes.

To IDS Flag: Remove IDS flag

Filter :: Remove filter

Reset filtering

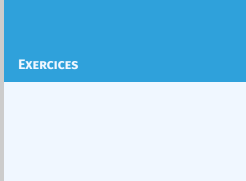
Filtering Label to remove: All filters



Fitlering data on which to apply on multiple modules

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

Exercices

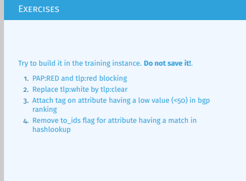


Exercices

Exercises

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

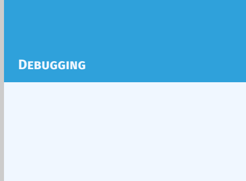
1. PAP:RED and tlp:red blocking
2. Replace tlp:white by tlp:clear
3. Attach tag on attribute having a low value (<50) in bgp ranking
4. Remove to_ids flag for attribute having a match in hashlookup


Exercises

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

1. PAP:RED and tlp:red blocking
2. Replace tlp:white by tlp:clear
3. Attach tag on attribute having a low value (<50) in bgp ranking
4. Remove to_ids flag for attribute having a match in hashlookup

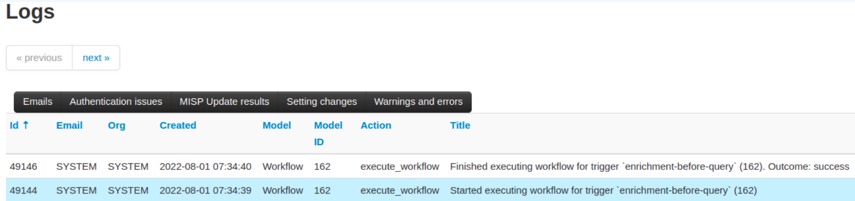
Debugging



Debugging

Debugging Workflows: Log Entries

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



Logs

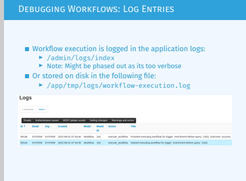
« previous | next »

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)

2024-04-15

Automation in MISP
└Debugging

└Debugging Workflows: Log Entries



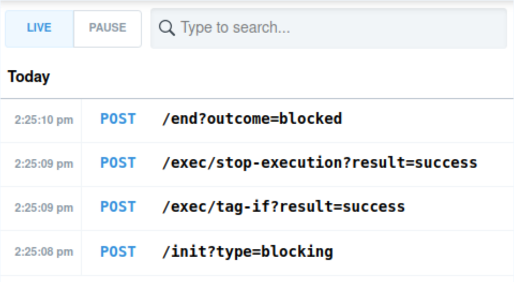
Debugging Workflows: Log Entries

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

Logs

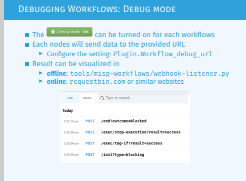
Debugging Workflows: Debug mode

- The Debug Mode: On 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



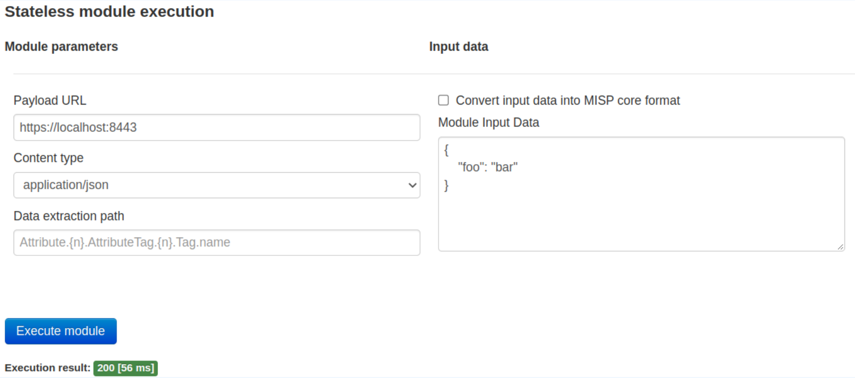
Today

```
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 Workflows: Debug mode
The Debug Mode: On 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

Debugging modules: Stateless execution

- Test custom modules with custom input



Stateless module execution

Module parameters

Payload URL

https://localhost:8443

Content type

application/json

Data extraction path

Attribute.{n}.AttributeTag.{n}.Tag.name

Input data

□ Convert input data into MISP core format

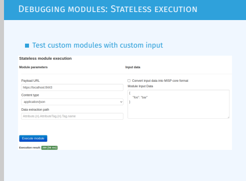
Module Input Data

```
{
    "foo": "bar"
}
```

Execution result: 200 [56 ms]

2024-04-15

Automation in MISP
└Debugging
└Debugging modules: Stateless execution

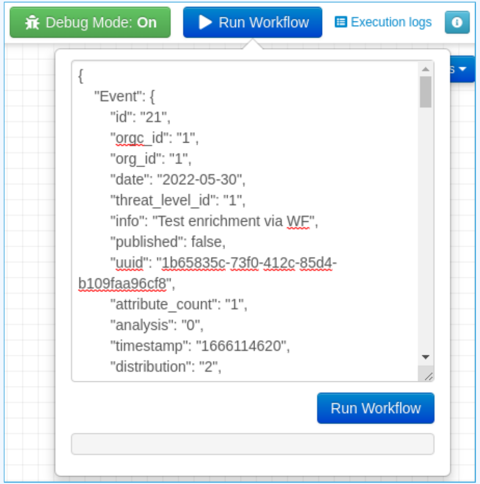


Debugging modules: Stateless execution

- Test custom modules with custom input

Debugging modules: Re-running workflows

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

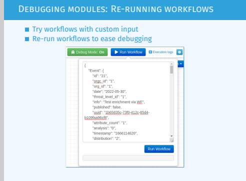


```
{
    "Event": {
        "id": "21",
        "orgc_id": "1",
        "org_id": "1",
        "date": "2022-05-30",
        "threat_level_id": "1",
        "info": "Test enrichment via WF",
        "published": false,
        "uuid": "1b65835c-73f0-412c-85d4-
b109faa96cf8",
        "attribute_count": "1",
        "analysis": "0",
        "timestamp": "1666114620",
        "distribution": "2",
```

2024-04-15

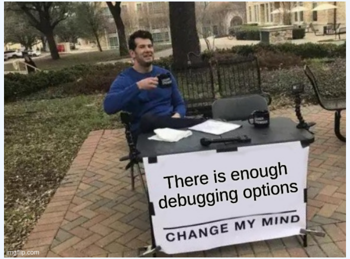
Automation in MISP
└Debugging

└Debugging modules: Re-running workflows


Debugging modules: Re-running workflows
- Try workflows with custom input
- Re-run workflows to ease debugging

Debugging options

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



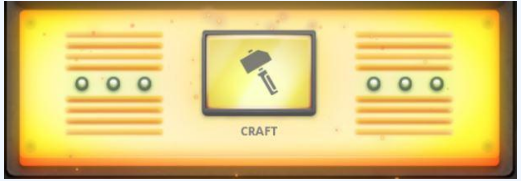
2024-04-15

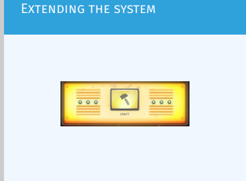
Automation in MISP
└Debugging
└Debugging options

Debugging options

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

Extending the system



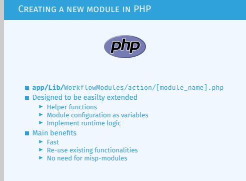

Extending the system

Creating a new module in PHP

- **app/Lib/**WorkflowModules/action/[module_name].php
- Designed to be easilty extended
  - Helper functions
  - Module configuration as variables
  - Implement runtime logic
- Main benefits
  - Fast
  - Re-use existing functionalities
  - No need for misp-modules

2024-04-15

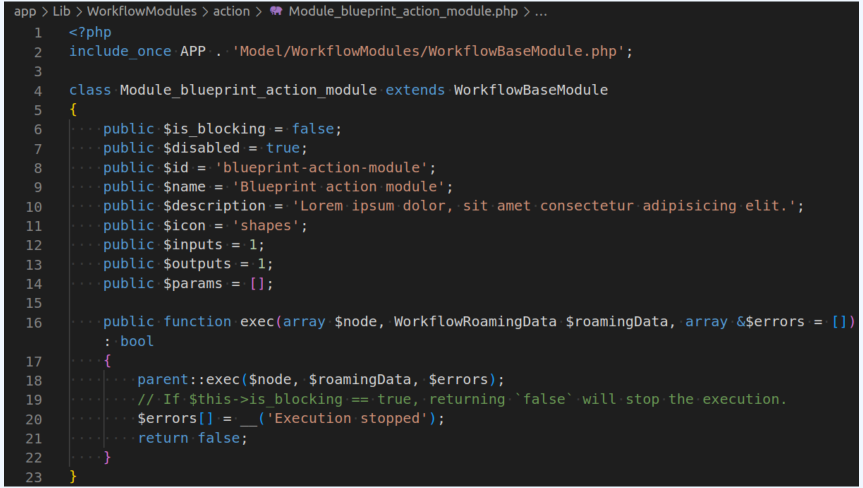
Automation in MISP
└Debugging
└Creating a new module in PHP



Creating a new module in PHP

- app/Lib/WorkflowModules/action/[module_name].php
- Designed to be easilty extended
  - Helper functions
  - Module configuration as variables
  - Implement runtime logic
- Main benefits
  - Fast
  - Re-use existing functionalities
  - No need for misp-modules

Creating a new module in PHP



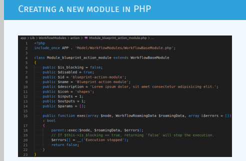
app > Lib > WorkflowModules > action > Module_blueprint_action_module.php > ...

```php
<?php
include_once APP . 'Model/WorkflowModules/WorkflowBaseModule.php';

class Module_blueprint_action_module extends WorkflowBaseModule
{
    public $is_blocking = false;
    public $disabled = true;
    public $id = 'blueprint-action-module';
    public $name = 'Blueprint action module';
    public $description = 'Lorem ipsum dolor, sit amet consectetur adipisicing elit.';
    public $icon = 'shapes';
    public $inputs = 1;
    public $outputs = 1;
    public $params = [];

    public function exec(array $node, WorkflowRoamingData $roamingData, array &$errors = []): bool
    {
        parent::exec($node, $roamingData, $errors);
        // If $this->is_blocking == true, returning `false` will stop the execution.
        $errors[] = __('Execution stopped');
        return false;
    }
}
```

2024-04-15

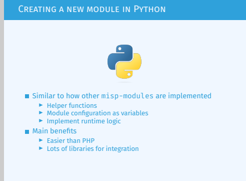
Automation in MISP
└Debugging

└Creating a new module in PHP



Creating a new module in Python

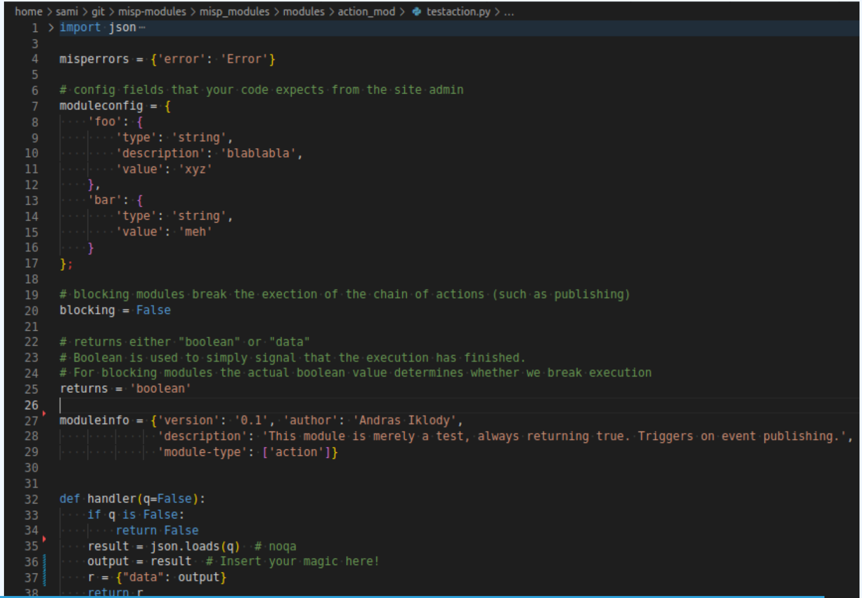
- Similar to how other `misp-modules` are implemented
  - Helper functions
  - Module configuration as variables
  - Implement runtime logic
- Main benefits
  - Easier than PHP
  - Lots of libraries for integration



Creating a new module in Python

- Similar to how other `misp-modules` are implemented
  - Helper functions
  - Module configuration as variables
  - Implement runtime logic
- Main benefits
  - Easier than PHP
  - Lots of libraries for integration

Creating a new module in Python

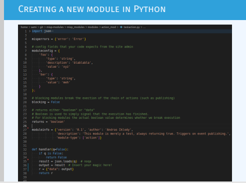


home > sami > git > misp-modules > misp_modules > modules > action_mod > testaction.py > ...

```
import json…

misperrors = {'error': 'Error'}

# config fields that your code expects from the site admin
moduleconfig = {
    'foo': {
        'type': 'string',
        'description': 'blablabla',
        'value': 'xyz'
    },
    'bar': {
        'type': 'string',
        'value': 'meh'
    }
};

# blocking modules break the exection of the chain of actions (such as publishing)
blocking = False

# returns either "boolean" or "data"
# Boolean is used to simply signal that the execution has finished.
# For blocking modules the actual boolean value determines whether we break execution
returns = 'boolean'

moduleinfo = {'version': '0.1', 'author': 'Andras Iklody',
              'description': 'This module is merely a test, always returning true. Triggers on event publishing.',
              'module-type': ['action']}


def handler(q=False):
    if q is False:
        return False
    result = json.loads(q)  # noqa
    output = result  # Insert your magic here!
    r = {"data": output}
    return r
```

2024-04-15

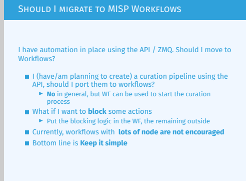
Automation in MISP
└Debugging
└Creating a new module in Python



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**
- Bottom line is **Keep it simple**



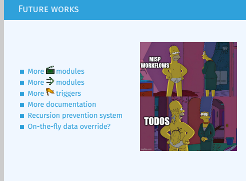
Should I migrate to MISP Workflows

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**
- Bottom line is **Keep it simple**

Future works

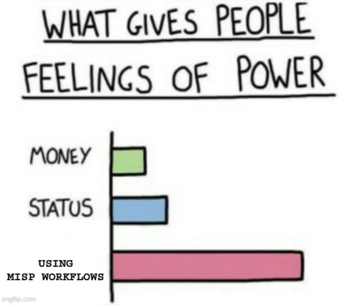
- More 🎬 modules
- More ⇛ modules
- More 🚩 triggers
- More documentation
- Recursion prevention system
- On-the-fly data override?

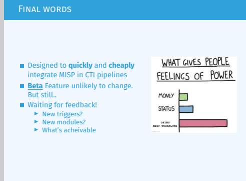

Future works

- More 🎬 modules
- More ⇛ modules
- More 🚩 triggers
- More documentation
- Recursion prevention system
- On-the-fly data override?

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?
  - What's acheivable




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?
  - What's acheivable