AN INTRODUCTION TO WORKFLOWS IN MISP

MISP - THREAT SHARING

CIRCL / TEAM MISP PROJECT

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

13TH ENISA-EC3 WORKSHOP



An Introduction to Workflows in MISP

8

2024-10

AN INTRODUCTION TO WORKFLOWS IN MISP MISP - THREAT SHARING



CONTENT OF THE PRESENTATION

- MISP Workflows fundamentals
- Getting started
- Design of the system & how it can be extended



An Introduction to Workflows in MISP

Content of the presentation



MISP Workflows fundamentals
 Getting started
 Design of the system & how it can be extended



3

2024

WHAT PROBLEMS ARE WE TRYING TO TACKLE

An Introduction to Workflows in MISP

└─What problems are we trying to tackle



Send notifications in a chat rooms
 And much much more.

Workshop organized by the Canadian Cyber Cente



Initial idea came during GeekWeek7.5¹

Needs:

- Prevent default MISP behaviors
- Hook specific actions to run callbacks

Use-cases:

- Prevent publication of events not meeting some criterias
- Prevent querying thrid-party services (e.g. virustotal) with sensitive information
- Send notifications in a chat rooms
- And much much more..

¹Workshop organized by the Canadian Cyber Center

9

An Introduction to Workflows in MISP

2024-10-02

WORKFLOW - FUNDAMENTALS

WORKFLOW - FUNDAMENTALS

SIMPLISTIC OVERVIEW OF A WORKFLOW IN ACTION

An Introduction to Workflows in MISP

8

0

2024-

Simplistic overview of a Workflow in action

- 1. An action happens in MISP
- 2. If there is an enabled Workflow for that action, run it
- 3. If all went fine, MISP continue to perform the action
 - r operation can potentiatly be cancelled by block dules

- 1. An **action** happens in MISP
- 2. If there is an **enabled** Workflow for that **action**, run it
- 3. If all went fine, MISP continue to perform the action
 - The operation can potentially be cancelled by blocking modules

TERMINOLOGY

- workflow: Sequence of all operations (nodes) to be executed. Basically the whole graph.
- **execution path**: A path composed of nodes
- **trigger**: Starting point of a workflow. Triggers are called when specific actions happen in MISP
 - ► A trigger can only have one workflow and vice-versa



An Introduction to Workflows in MISP

└─ Terminology

8

2024-10

 workflow: Sequence of all operations (nodes) to be executed. Basically the whole graph. execution path. A path composed of nodes trigger: Starting point of a workflow. Triggers are called when specific actions happen in MISP A trigger can only have one workflow and vice-versa



WORKFLOW EXECUTION PROCESS

Typical execution process:

- 1. An action happens in MISP
- 2. The workflow associated to the trigger is ran
- 3. Execution result?
 - success: Continue the action
 - failure | blocked: Cancel the action

Example for Event publish:

- 1. An Event is about to be published
- 2. MISP executes the workflow listening to the event-publish trigger
 - **success**: Continue the publishing action
 - failure | blocked: Stop publishing and log the reason

An Introduction to Workflows in MISP

└─Workflow execution process

WORKFLOW EXECUTION PROCESS

Pypical execution proces

- 1. An action happens in MISP
- 2. The workflow associated to the trigger is ra
- 3. Execution result?
- success: Continue the action
 failure | blocked: Cancel the action

Example for Event publis

- An Event is about to be published
 MISP executes the workflow listening to th
- event-publish trigger
 - success: Continue the publishing action
 failure | blocked: Stop publishing and log the reas

8

0

BLOCKING AND NON-BLOCKING WORKFLOWS

An Introduction to Workflows in MISP

Blocking and non-blocking Workflows

Currently 2 types of workflow

- Blocking: Completion of the action can be prevented
 If a blocking module blocks the action
- If a blocking module raises an exception
- Non-blocking: Workflow execution outcome has no impact Blocking mediates can still stop the execution

Currently 2 types of workflows:

- **Blocking**: Completion of the action can be prevented
 - ► If a **blocking module** blocks the action
 - ► If a **blocking module** raises an exception
- Non-blocking: Workflow execution outcome has no impact
 - **Blocking modules** can still stop the execution

8

0

EXECUTION CONTEXT

An Introduction to Workflows in MISP 은 └─Workflow - Fundamentals

Execution context

Workflows can be triggered by any users

■ Workflows can be triggered by actions done via the UI or API

- However, the user for which the workflow executes has:
 - The site-admin permission
 - Is from the MISP.host_org_id
- Ensures data is processed regardless of ownership and access: no ACL

0

CLASSES OF WORKFLOW MODULES

Actions

3 classes of modules

- **action**: Allow to executes functions, callbacks or scripts
 - Can stop execution
 - e.g. Webhook, block the execution, perform enrichments, ...
- **logic**: Allow to redirect the execution flow.
 - IF condition, fork the blocking execution into a non-blocking one, ...
- **blueprint**: Allow to reuse composition of modules
 - Can save subworkflows and its module's configuration

An Introduction to Workflows in MISP

Classes of Workflow modules

	Actions		A Discorts	
3 classes o				
action	Allow to exe		ions, callbacks or	r scri
action	Allow to exer n stop execution	in .		
■ action ► Ca ► e.	Allow to exer n stop execution	in lick the exec	ution, perform enr	

Can save subworkflows and its module's configuration

8

SOURCES OF WORKFLOW MODULES

3 sources of action modules

Built-in **default** modules

- Part of the MISP codebase
- app/Model/WorkflowModules/action/[module_name].php

User-defined custom modules

- ► Written in PHP
- Can extend existing default modules
- Can use MISP's built-in functionalities (restsearch, enrichment, push to zmq, ...)
- **Faster and easier to implement new complex behaviors**
- app/Lib/WorkflowModules/action/[module_name].php

An Introduction to Workflows in MISP

8

0

2024-

└─Sources of Workflow modules

sources of action modules Built-in default modules

- Built-in default modules
- Part of the MISP codebase
 app/Hodel/WorkflowModules/action/[module_na
- User-defined custom modules
 - Written in PHP
 - Can extend existing default modules
 Can use MISP's built-in functionalities (rest
 - enrichment, push to zmq, ...)

Sources of Workflow modules

3 sources of action modules

- Modules from the enrichment service
 - Default and custom modules
 - ► From the *misp-module*
 - ► Written in Python
 - Can use any python libraries
 - New misp-module module type: action
 - \rightarrow Both the PHP and Python systems are plug-and-play

An Introduction to Workflows in MISP

└─Sources of Workflow modules



Both the PHP and Python systems are plug-and-play

8

TRIGGERS CURRENTLY AVAILABLE

An Introduction to Workflows in MISP

-02

2024-10

└─Triggers currently available

Current			ers can be hoo	ked.	gblo	king	t	_	
	-	800	Particular Scalar des antimiser sonant						••••
· beinen bien	-		The tage is a fee (and only a propagate in party of a rate (a fee)	95			10+1+5.5		
a new second	-		The type is solve the article that has seen in						••••
à branhadan			The same is a first part of the a staff Concerns.						
a Red for her	-	000	Participants advender an High the feet recent in the sectors						
	-		The same is a first of the data and a second to be defined as				******		••••
A the Marine	-	•	The tage is added to a set for the section in the section of the s						
	-		The lage is also provide a set of some						••••

Currently 8 triggers can be hooked. 3 being **blocking**.

A thribute Atter Save articular This trigger is called after an Attribute has been 58 × 160 2022-07-29 0638:11 IV-II K Enrichtmert Before ofters Imager is called after an Attribute has been saved in 611 ✓ 160 2022-07-29 0632:32 IV-III K Enrichtmert Before ofters Imager is called after an Attribute has been saved in 611 ✓ 162 2022-07-29 0632:32 IV-IIII B Event After Save event rendium O This trigger is called after an Event has been saved in 111 × ✓ 160 2022-07-29 0632:32 IV-IIIII S. Event After Save event rendium O This trigger is called after an Event has been saved in 11 × ✓ 180 2022-07-29 0637:23 IV-IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII		-	00			-	0		_		
* Enrichment Before other saved in the database * In this toger is called just before a query against the 941 ✓ 102 2022-07-20 08.25.25 * In this toger is called just before a query against the 941 ✓ 102 2022-07-20 08.25.25 * In this toger is called just before a query against the 941 ✓ 102 2022-07-20 08.25.25 * In this toger is called just before a query against the 941 ✓ 102 2022-07-20 08.25.25 * In this toger is called just before a MSP Event starts 11 × ✓ 175 2022-07-20 12.14.10 * In this toger is called just before a MSP Event starts 1 ✓ ✓ 180 2022-07-20 12.14.10 * In this toger is called just before a MSP Event starts 1 ✓ ✓ 181 2022-07-20 13.59.37 × In this toger is called after an Object has been saved in 36 × × 176 2022-07-20 13.59.37 × In the database Post After Save post in the database 26 × × 176 2022-07-20 13.59.37 × In the database Lyser After Save user text after a Post has been saved in 36 × × 176 2022-07-28 13.59.51 In the data	Trigger name	Scope	Trigger overhead	Description	Run counter	Blocking Workflow	MISP Core format	Workflow ID	Last Update	Enabled	Actions
Query encloment service is done Event After Save event medianical after an Degic Lailed after an Degic has been saved in 11 × <	Attribute After Save	attribute	high 😧		58	×	*	160	2022-07-29 06:58:11	*	∎⋪⊟©
Be database Image: Control of the database Image: Control of the publishing process Image: Control of the publishing	* Enrichment Before Query	others	low		841	~	*	162	2022-07-29 08:32:32	*	∎⋪∎๏
We object After Save object This trigger is called after an Object has been saved in 36 × ✓ 161 2022-07-28 1359-37 × ✓ 0 P Post After Save post Imager is called after an Object has been saved in 36 × × 176 2022-07-28 1359-51 ✓ ● ● L User After Save user Imager is called after an user has been saved in 35 × × 159 2022-07-28 1359-51 ● <t< td=""><td>Event After Save</td><td>event</td><td>medium 😧</td><td></td><td>11</td><td>×</td><td>*</td><td>175</td><td>2022-07-29 08:37:23</td><td>*</td><td>∎∲⊟∅</td></t<>	Event After Save	event	medium 😧		11	×	*	175	2022-07-29 08:37:23	*	∎∲⊟∅
Post Atter Save post Im de diablase L User Atter Save user Im de diablase L User Before Save user Im de diablase L User Save User X Im de diablase L User Save User X Im de diablase L User Save User <	1 Event Publish	event	low	00 7	1	~	~	180	2022-07-29 12:14:10	*	∎∲≣∅
be database b User After Save user to sale of the database b User After Save user to sale of the database b User Before Save user to sale of the database b User Before Save user to sale of the database b User Before Save user to sale of the database	& Object After Save	object	high 😧		35	×	*	161	2022-07-28 13:59:37	×	▶⋪∎∅
the database & User Before Save user I Ibm This trigger is called just before a user is save in the 42 × × 158 2022-07-28 14:00.32 × ■ $\phi \blacksquare \phi$	Post After Save	post	low		36	×	×	176	2022-07-28 13:59:51	*	∎⋪∎∅
	🎒 User After Save	user	low		55	×	×	159	2022-07-28 14:00:03	*	∎⋪∎ଡ଼
	≗ + User Before Save	user	low	55 7	42	~	×	158	2022-07-28 14:00:32	*	∎∲≣∅

An Introduction to Workflows in MISP

WORKFLOW - GETTING STARTED

WORKFLOW - GETTING STARTED

GETTING STARTED WITH WORKFLOWS (1)

Review MISP settings:

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

Overview MI	SP settings (20 🛕)	Encryption set	tings (7 🛕) 🛛 Prox	y settings (5)	Security settings (8 \land)	Plugin settings (465 🕰)	SimpleBackgroundJobs settings (1	1 🗛) Di
Enrichment							Filter the table(s) below	
Import								
Export								
Action								
Critical	Plugin.Action_servic	es_enable	true	Enabl	e/disable the action service	15		
Recommended	Plugin.Action_servic	es_url	http://host.docker.i	nternal The u	rl used to access the action	services. By default, it is a	ccessible at http://127.0.0.1:6666	
Recommended	Plugin.Action_servic	es_port	6677	The p	ort used to access the activ	on services. By detault, it is	accessible at 127.0.0.1:6666	
Recommended	Plugin.Action_timeo	ut	10	Set a	timeout for the action servi	ces		Value not set.

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

Overview	MISP settings (20 🛦)	Encryption settings (7 🛦)	Praxy settings (5)	Security settings (8 🛕)	Plugin settings (465 🛕)	SimpleBackgroundJobs settings (11 \Lambda)	Diagno
Enrichmen						Filter the table(s) below	
Import							
Export							
Action							
Cortex							
Sightings							
Workflow							
Recomme	nded Plugin.Workflow_er	nable	true Enable	aldisable workflow feature			

An Introduction to Workflows in MISP

8

2024

 \Box Getting started with workflows (1)

1. Mai 2. Mai	MISP settings: ke sure MISP.background_jobs is turned o ke sure workers are up-and-running and healt n the setting Plugin.Workflow_enable on
	And Alexandra Supervised Advanced Annual Annua
	The Party and the Party of the
	tional:misp-module]Turn the setting ugin.Action_services_enable on
	ten transfe Tenners & Long Tenners Second Statements

GETTING STARTED WITH WORKFLOWS (2)

An Introduction to Workflows in MISP

8

2024-10

Getting started with workflows (2)

GETTING STARTED WITH WORKFLOWS (2)

f you wish to use action modules from misp-module, make unre to have: The latest update of misp-module > There should be an action_mod module type in

misp-modules/misp_modules/modules

his command should show all "action" modules
wrl == http://siz.co.six6666/modules | /
'.[] | select (.meta."module-type"[] | contains("action")) |
me: .mane, version: .meta.version]'

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

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

GETTING STARTED WITH WORKFLOWS (3)

An Introduction to Workflows in MISP

Getting started with workflows (3)

- 1. Go to the list of modules
- Administration > Workflows > List Modul
- Make sure default modules are loaded
- footional:misp-module] Make sure misp-module module:
- are loaded

1. Go to the list of modules

- Administration > Workflows > List Modules
- or /workflows/moduleIndex
- 2. Make sure **default** modules are loaded
- 3. [optional:misp-module] Make sure **misp-module** modules are loaded

8

0

CREATING A WORKFLOW WITH THE EDITOR

- 1. Go to the list of triggers Administration > Workflows
- 2. Enable and edit a trigger from the list
- 3. Drag an action module from the side panel to the canvas
- From the trigger output, drag an arrow into the action's input (left side)
- 5. Execute the action that would run the trigger and observe the effect!

Trigger name	Scope	Trigger overhead	Description	Run counter	Blocking Workflow	MISP Core format	Workflow ID	Last Update	Enabled	Actions
C Attribute After Save	atribute	high 0	This trigger is called after an Attribute has been saved in the database	58	×	×	160	2022-07-29 06:58:11	*	•••••
* Enrichment Before Query	others	kow	This trigger is called just before a query against the enrichment service is done	841	*	~	162	2022-07-29 08:32:32	*	∎∲≣⊚
Event After Save	event	medium	This trigger is called after an Event has been saved in the database	11	×	×	175	2022-07-29 08:37:23	*	■\$0
1 Event Publish	event	kow	This trigger is called just before a MISP Event starts the publishing process	1	*	~	180	2022-07-29 12:14:10	*	■\$
& Object After Save	object	high 0	This trigger is called after an Object has been saved in the database	35	×	~	161	2022-07-28 13:59:37	×	▶ \$ ₽∎€
Post After Save	post	kow	This trigger is called after a Post has been saved in the database	36	×	×	176	2022-07-28 13:59:51	×	•\$\$
🎝 User After Save	user	law	This trigger is called after a user has been saved in the database	55	×	×	159	2022-07-28 14:00:03	*	∎¢∎⊛
A* User Before Save	user	low	This trigger is called just before a user is save in the database	42	×	×	150	2022-07-28 14:00:32	*	•\$



An Introduction to Workflows in MISP

8

0

2024

Creating a workflow with the editor

inp	st (left side)	ger output,	dra	ig a	n arr	Św	he ac
	effect!	ion that wo	ula		the		
	Statement and state						-
	general 11 🗰						
	second of \$50						1100
		Contraction of the second second					

WORKING WITH THE EDITOR

Operations not allowed:

- Execution loop are not authorized
 - Current caveat: If an action re-run the workflow in any way



An Introduction to Workflows in MISP

└─Working with the editor

WORKING WITH THE EDITOR

Operations not allowed: Execution loop are not authorized Current caveat: If an action re-run the workflow in any way



16

30

-02

WORKING WITH THE EDITOR

An Introduction to Workflows in MISP

└─Working with the editor

WORKING WITH THE EDITOR

Operations not allowed: Multiple connections from the same output Forentine order not suscented and confining for users.



Operations not allowed:

- Multiple connections from the same output
 - **•** Execution order not guaranted and confusing for users



-02

WORKING WITH THE EDITOR

An Introduction to Workflows in MISP

└─Working with the editor



Derations showing a warning: Blocking modules after a concurrent tasks module Blocking modules in a non-blocking workflow

1 free banks	 W Concerned Task	 S line security O	
	 And the second s	 Scaling is an including one	

Operations showing a warning:

- Blocking modules after a concurrent tasks module
- **Blocking** modules in a **non-blocking** workflow

		🗙 Concurrent Task 🛛 🔤	Stop execution O
, Event Publish	O Blocking 📖 🖗	Allow breaking the execution process and running concurrent tasks. You can connect multiple nodes the "concurrent" output.	Essentially stops the execution for blocking workflows. Do nothing for non-blocking ones

10-02

2024

WORKFLOW BLUEPRINTS

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

Debugging webhook	v1656059209
Default: ×	
Blueprint Content: 1 node	
Webhook module pre-configured for deb purposes	bugging

Blueprints origins:

- From the "official" misp-workflow-blueprints repository
- 2. Created or imported by users

An Introduction to Workflows in MISP

8

2024-10

└─Workflow blueprints

Debugging webhook
Defail: #
Excepter Content & sode
& L
triditook medule pre-configured for debugging purposes

WORKFLOW BLUEPRINTS: CREATE

Select one or more modules to be saved as blueprint then click on the save blueprint button



An Introduction to Workflows in MISP

10-02

2024

└─Workflow blueprints: Create

WORKFLOW BLUEPRINTS: CREATE

Select one or more modules to be saved as blueprint then click on the save blueprint button



20

HASH PATH FILTERING

Some modules have the possibility to filter or check conditions using CakePHP's path expression.

```
1 $path_expression = '{n}[name=fred].id';
2 $users = [
3 {'id': 123, 'name': 'fred', 'surname': 'bloggs'},
4 {'id': 245, 'name': 'fred', 'surname': 'smith'},
5 {'id': 356, 'name': 'joe', 'surname': 'smith'},
6 ];
7 $ids = Hash::extract($users, $path_expression);
8 // => $ids will be [123, 245]
```

Value	
tlp:red	
Operator	
In	```
Hash path	
Attribute.{n}.Tag	

```
2024-10-02
```

An Introduction to Workflows in MISP

└─ Hash path filtering

	odules have the possibility to filter or check ons using CakePHP's path expression.
2 Susers + [3 ['id': 4 ['id': 5 ['id': 6]; 7 Sids + Hast	<pre>sion = "[n][nume=fred].id"; zz, "nume": "fred", "uurnamesi: "bioggst], st, "nume": "red", "uurnamesi: "smith"], 56, "nume": "jee", "numamesi: "smith"], estract(Suures, Spath_supression); tit be [zzz, zdz]</pre>
	P # :: densets

MODULE FILTERING

Some action modules accept filtering conditions
 E.g. the enrich-event module will only perform the enrichment on Attributes having a tlp:white Tag

Module Filtering		×
Element selector		
Attribute.{n}		
Value		
tlp:white		
Operator		
In		~
Hash Path		
AttributeTag.{n}.Tag.name		
	Save	Close

2024-10-02

An Introduction to Workflows in MISP

└─Module filtering

Notule Filtering	
Deners alledar	
All board (r)	
Value	
Sparter	
(mentry	
in the second se	
1000 840	
Amban han	

DATA FORMAT IN WORKFLOWS

🛓 Event Publish 🛛 🖸 Blocking 🛄 🖕

- All triggers will inject data in a workflow
- In some cases, there is no format (e.g. User after-save)
- In others, the format is **compliant with the MISP Core format**
- In addition to the RFC, the passed data has additional properties
 - Attributes are **always encapsulated** in the Event or Object
 - Additional key _AttributeFlattened
 - Additional key _allTags
 - Additional key inherited for Tags

An Introduction to Workflows in MISP

8

ģ

2024-

L Data format in Workflows





LOGIC MODULE: CONCURRENT TASK

- Special type of logic module allowing multiple connections
 Allows breaking the execution flow into a concurrent tasks to be executed later on by a background worker
- As a side effect, blocking modules cannot cancel ongoing operations



An Introduction to Workflows in MISP

8

ġ

2024-

Logic module: Concurrent Task

LOGIC MODULE: CONCURRENT TAS

 Special type of logic module allowing multiple connections
 Allows breaking the execution flow into a concurrent tasks to be executed later on by a background worker
 As a side effect, blocking modules cannot cancel ongoing onerations



DEBUGGING WORKFLOWS: LOG ENTRIES

Workflow execution is logged in the application logs:

- /admin/logs/index
- Or stored on disk in the following file:
 - /app/tmp/logs/workflow-execution.log
- Use the webhook-listener.py tool
 - /app/tools/misp-workflows/webhook-listener.py

Logs		»					
Email	s Authen	ication issue	es MISP Update res	ults Setti	ng changes	Warnings and e	rrors
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)

An Introduction to Workflows in MISP

8

2024-10

Lebugging Workflows: Log Entries

	Wor	rkfle	w exect	ution	i is lo	bezed	in the application logs:
		/a	imin/lo	igs/		x	
	Or :	ston	ed on di	sk ir	the	follow	ving file:
		/a1	o/tmo/	100	s/wo	rkfle	w-execution.log
			webho				
							.py tool 'lows/webhook-listener.
	•						
Log	•						
Log	•	/aj					
Log	•	/aj					
Log	•	/aj					
Log		/a		s/8	isp-	workf	

DEBUGGING WORKFLOWS: DEBUG MODE



n 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

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

An Introduction to Workflows in MISP Workflow - Getting started

8

0 2024-

-Debugging Workflows: Debug mode

NG WOI	RKF	LOWS: DEBUG MODE	
Debug Model	De:		
			nmows
figure th	ie se	tting: Plugin.Workflow_deb	ig_url
an be vi		lized in	
ine: requ	uest	bin.com or similar websites	istener.py
Teday			
235-23pm	POST	/and heatscame-blacked	
10.000	POST	/em/step-emotion/real-transms	
		/ene/ship-enerties/vesitionness /ene/tap-tifresitionerss	
	teogree des will figure th tan be v ine: too ine: req ney ney ney	teogram te des will sen rigure the ser ine: tools/s ine: roquest ne: roquest ne: roquest	NG WORKFLOWS: DEBUG MODE

An Introduction to Workflows in MISP

LEARNING BY EXAMPLES

LEARNING BY EXAMPLES

WORKFLOW EXAMPLE 1

An Introduction to Workflows in MISP $\begin{subarray}{c} \mathsf{S} \\ \begin{subarray}{c} \mathsf{Learning} \end{subarray}$ by examples

└─Workflow example 1

2024



 The Event-Publish trigger uses the MISP core format
 The LF::Tag module checks if at least one of the Attribute has the tlp:white tag
 If it does, the Push-to-ZMQ module will be executed



- 1. The Event-Publish trigger uses the MISP core format
- 2. The IF::Tag module checks if at least one of the Attribute has the tlp:white tag
- 3. If it does, the Push-to-ZMQ module will be executed

WORKFLOW EXAMPLE 2

An Introduction to Workflows in MISP

Workflow example 2

2024-



If an event has the tlp:red tag or any of the attribute has it, the publish process will be cancelled



If an event has the tlp:red tag or any of the attribute has it, the publish process will be cancelled

An Introduction to Workflows in MISP

EXTENDING THE SYSTEM

EXTENDING THE SYSTEM

CREATING A NEW MODULE IN PHP



- app/Lib/WorkflowModules/action/[module_name].php
- Module configuration are defined as public variables
- The exec function has to be implemented.
 - ► If it returns **true**, execution will proceed
 - ► If it returns **false**
 - And the module is blocking, the execution will stop and the operation will be blocked

An Introduction to Workflows in MISP

8

2024-10

└─Creating a new module in PHP

app/Lib/storkflowHodules/action/[module_name].ph
 Module configuration are defined as public variables
 The exec function has to be implemented
 Fift returns true, execution will proceed
 Fift returns false
 A site module is blocking, the execution will step and the

29

CREATING A NEW MODULE IN PYTHON

	sami > git > misp-modules > misp_modules > modules > action_mod > 🚸 testaction.py >
3	
- 4	
5	
6	
7	moduleconfig = {
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	blocking = False
21	
22	
23	
24	
25	returns = 'boolean'
26	
27	moduleinfo = {'version': '0.1', 'author': 'Andras Iklody',
28	colordordorderightion': 'This module is merely a test, always returning true, Triggers on event publishing.'
29	<pre>'module-type': ['action']}</pre>
30	
31	
32	def-handler(g=False):
33	1f g is False:
34	
35	···· result ison.loads(q) ··#-noga
36	••••output = result • # Insert your magic here!
37	····r = {"data": output}
38	
39	
40	
41 >	def introspection():-
64	

 Module configuration are defined in the moduleinfo and moduleconfig variables
 The handler function has to be implemented.

Blocking logic is the same as other modules

An Introduction to Workflows in MISP

└─Creating a new module in Python



Exception of the section of the			
Exception of the section of the			
	the second se		
1 Service State			
The second secon			
A TOTAL CONTRACTOR) — 'E		
And a first start of the start	-		
	1		

 Module configuration are defined in the moduleinfo and moduleconfig variables
 The handler function has to be implemented.
 Blocking logic is the same as other modules

02