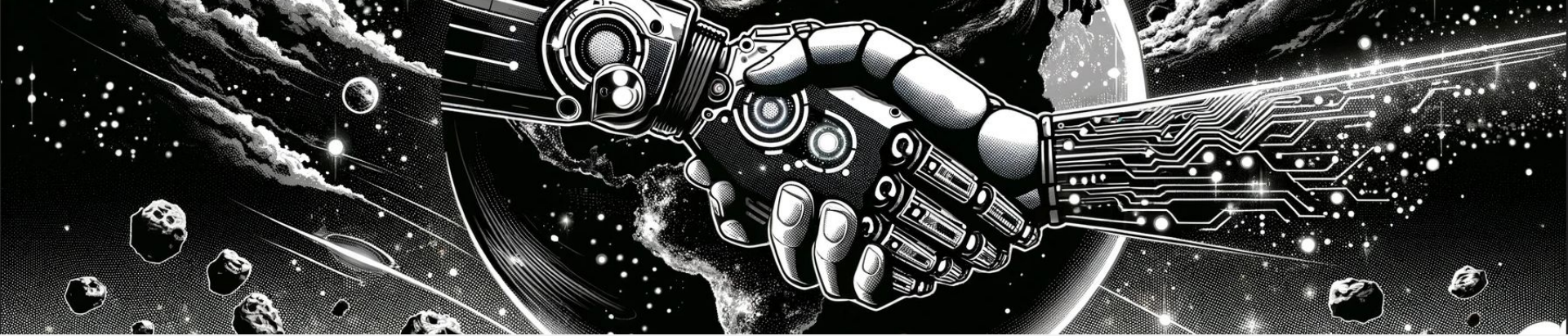


Integrating New Tools in Your Workflows Within Minutes in MISP

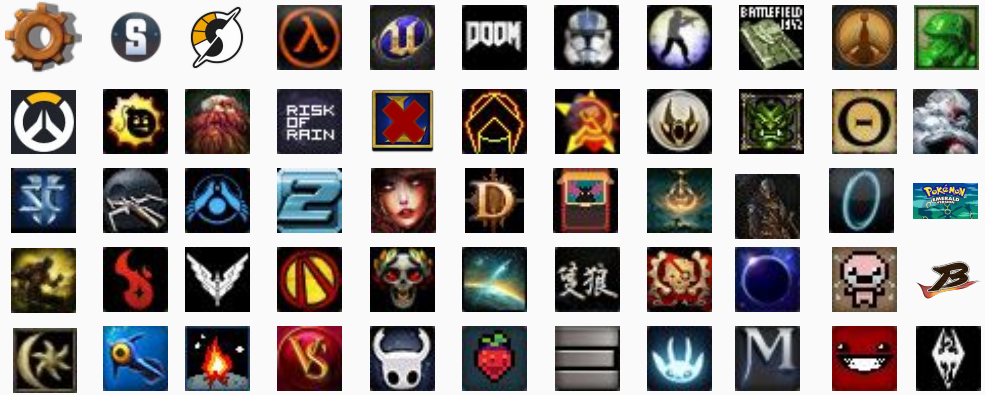
MISP
Threat Sharing

HACK.LU

Sami Mokaddem

- Working at CIRCL since 2016
- Part of the MISP-Project team
- 

- Love video games



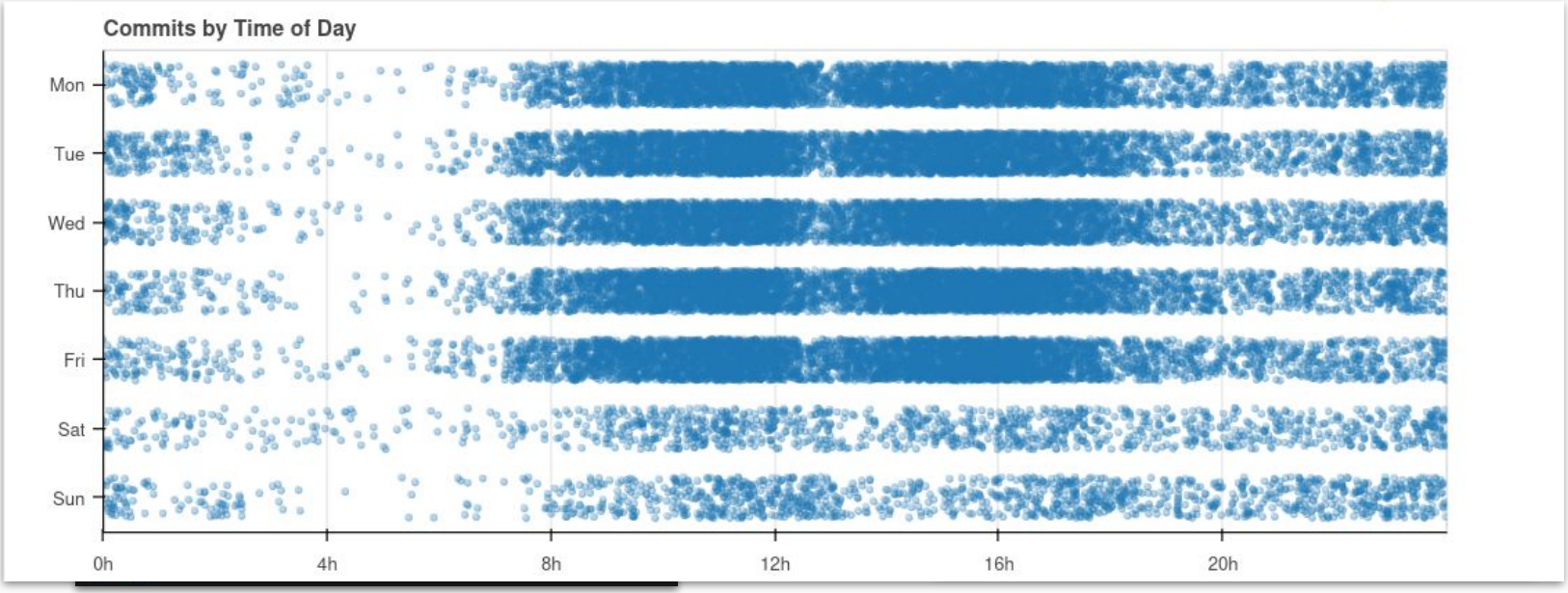
Building MISP Features with Music...

AIL Project: Secrets in Squares - QR Codes

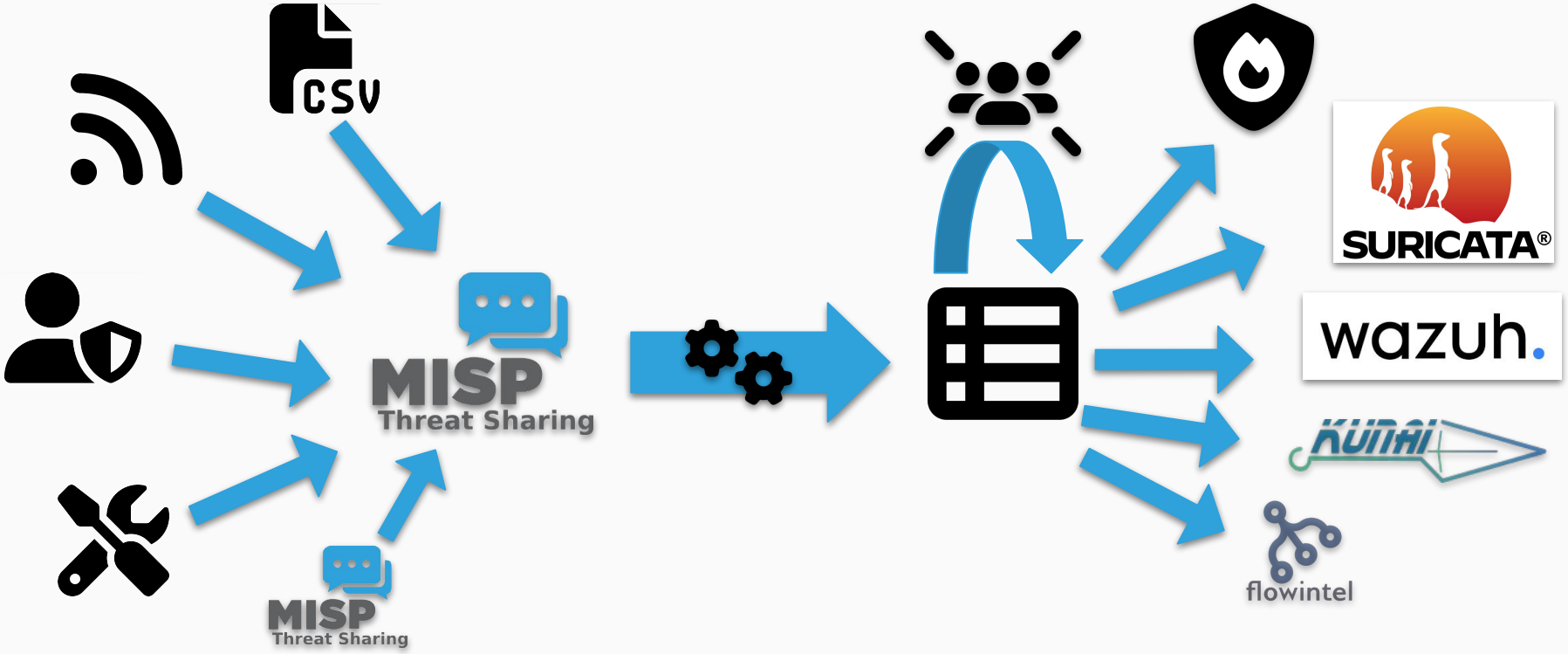
CIRCL
Computer Incident
Response Center
Luxembourg

Aurelien Thirion
aurelien.thirion@circl.lu

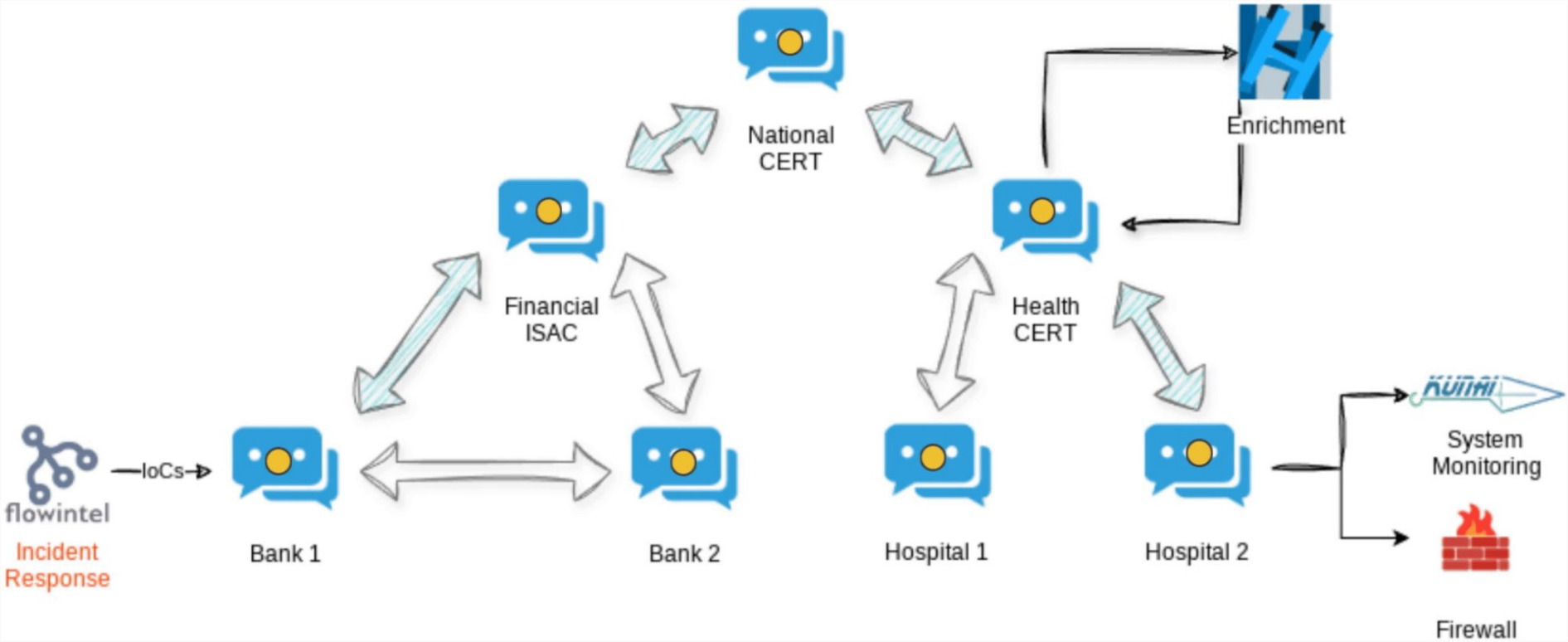
Threat Intelligence Sharing Platform - TISP



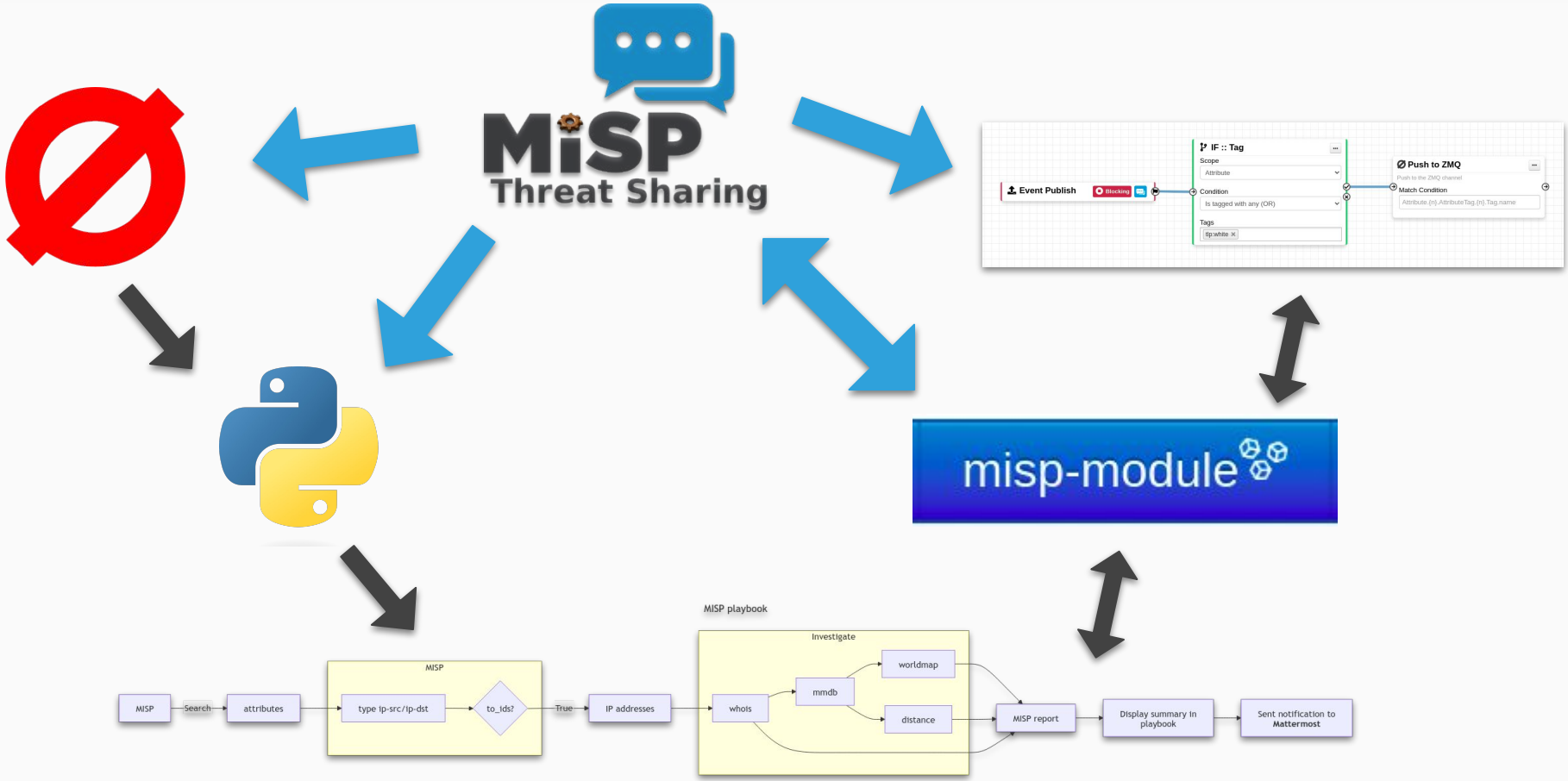
MISP? What can I use it for?



MISP Community & Information Sharing



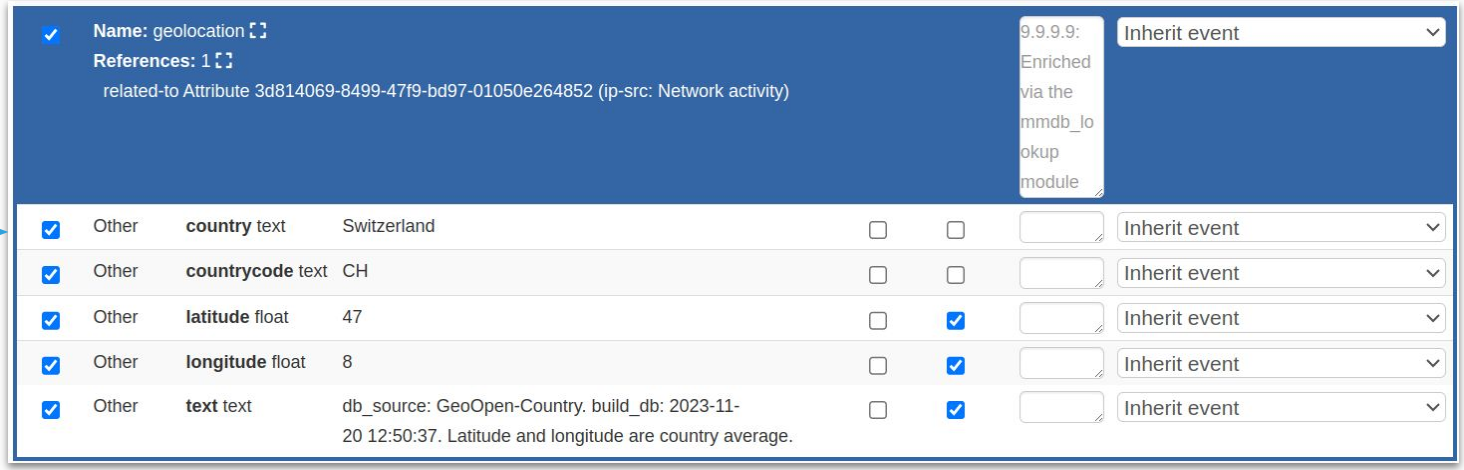
MISP Automation Ecosystem



misp-module

Category	Type	Value
Network activity	ip-src	9.9.9.9

give_me_geolocation.py



misp-module

```
root@sami-T14:/home/sami/git/misp-modules/misp_modules/modules/expansion# ls
abuseipdb.py                    extract_url_components.py       malwarebazaar.py                sophoslabs_intelix.py
apiosintds.py                   farsight_passivedns.py          mcafee_insights_enrich.py       sourcecache.py
apivoid.py                      geoip_asn.py                    mmdb_lookup.py                  stairwell.py
assemblyline_query.py           geoip_city.py                   module_misp_standard.py.skeleton stix2_pattern_syntax_validator.py
assemblyline_submit.py          geoip_country.py                module.py.skeleton              threatcrowd.py
backscatter_io.py               google_safe_browsing.py         mwdb.py                         threatfox.py
btc_scam_check.py               google_search.py                ocr_enrich.py                   threatminer.py
btc_steroids.py                 google_threat_intelligence.py   ods_enrich.py                   triage_submit.py
censys_enrich.py                greynoise.py                    odt_enrich.py                   trustar_enrich.py
circl_passivedns.py             hashdd.py                       onyphe_full.py                  urlhaus.py
circl_passivessl.py             hashlookup.py                   onyphe.py                       urlscan.py
clamav.py                       hibp.py                         otx.py                          variotdbs.py
cluster25_expand.py             html_to_markdown.py             passive_ssh.py                  virustotal_public.py
convert_markdown_to_pdf.py      hyasinsight.py                  passivetotal.py                 virustotal.py
countrycode.py                  __init__.py                     pdf_enrich.py                   virustotal_upload.py
cpe.py                          intel471.py                     pptx_enrich.py                  vmray_submit.py
crowdsec.py                     intelmq_eventdb.py.experimental __pycache__                     vmware_nsx.py
crowdstrike_falcon.py           ip2locationio.py                qintel_qsentry.py               vulndb.py
cuckoo_submit.py                ipasn.py                        qrcode.py                       vulnerability_lookup.py
custom_custom.py                ipinfo.py                       _ransomcoindb                   _vulnerability_parser
cve_advanced.py                 ipqs_fraud_and_risk_scoring.py  ransomcoindb.py                 vulners.py
cve.py                          iprep.py                        rbl.py                          vysion.py
cytomic_orion.py                jinja_template_rendering.py     recordedfuture.py               whoisfreaks.py
dbl_spamhaus.py                 joesandbox_query.py             reversedns.py                   whois.py
_dnsdb_query                    joesandbox_submit.py            securitytrails.py               wiki.py
dns.py                          lastline_query.py               shodan.py                       xforceexchange.py
docx_enrich.py                  lastline_submit.py              sigma_queries.py                xlsx_enrich.py
domaintools.py                  macaddress_io.py                sigma_syntax_validator.py       yara_query.py
eql.py                          macvendors.py                   sigmf_expand.py                 yara_syntax_validator.py
eupi.py                         malshare_upload.py              socialscan.py                   yeti.py
root@sami-T14:/home/sami/git/misp-modules/misp_modules/modules/expansion# ls | wc
    120     120    1926
root@sami-T14:/home/sami/git/misp-modules/misp_modules/modules/expansion#
```

MISP Workflow

Triggers

List the available triggers that can be listened to by workflows.
Missing a trigger? Feel free to open a Github issue!
Documentation and concepts

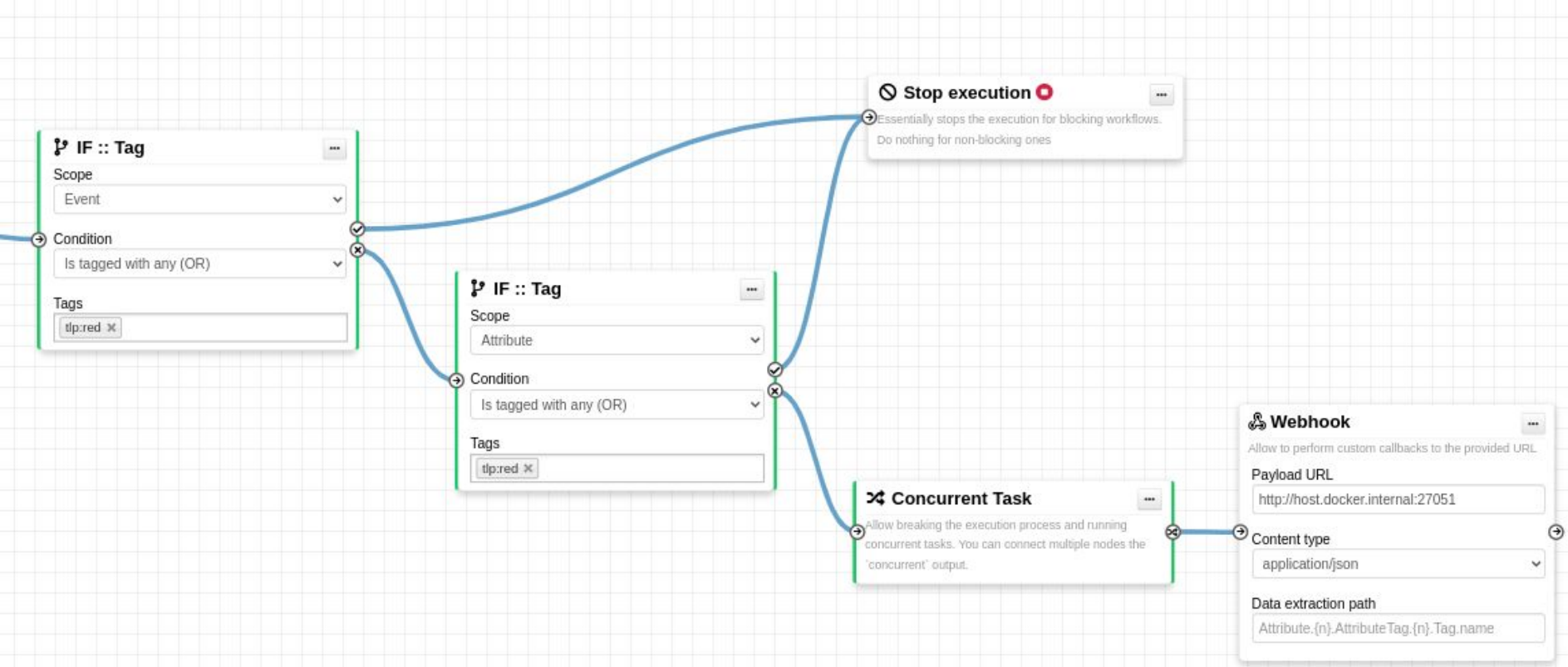
« previous | next »

All | attribute | event | event-report | log | object | others | post | shadow-attribute | sighting | tag | user | Blocking | Enabled | Disabled

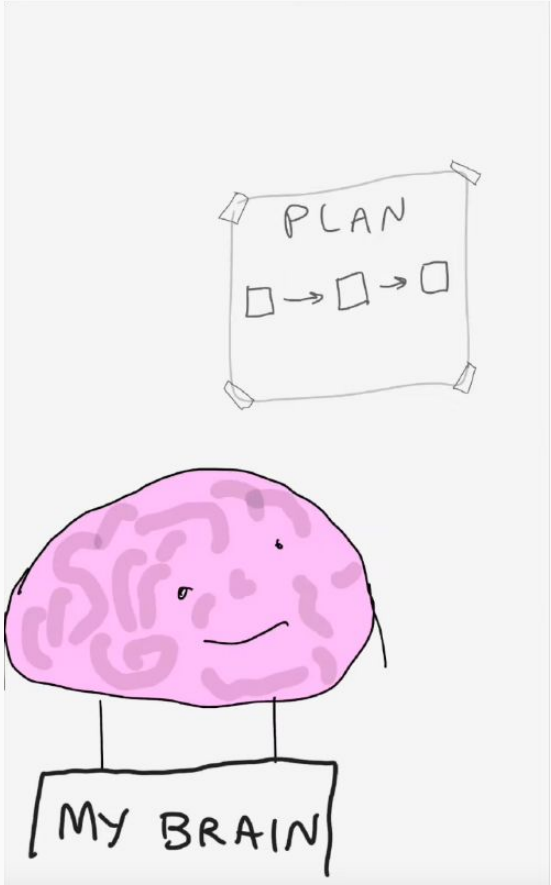
Trigger name	Scope	Trigger overhead	Description	Run counter	Blocking Workflow	MISP Core format	Workflow ID	Last Update	Debug enabled	Enabled	Actions
Attribute After Save	attribute	high	This trigger is called after an Attribute has been saved in the database	6075	×	✓	11	2024-10-18 09:03:37	☐	✓	
Enrichment Before Query	others	low	This trigger is called just before a query against the enrichment service is done		✓	✓				×	
Event After Save	event	high	This trigger is called after an Event or any of its elements has been saved in the database	2	×	✓	8	2024-10-07 10:18:53	☐	×	
Event After Save New	event	low	This trigger is called after a new Event has been saved in the database		×	✓				×	
Event After Save New From Pull	event	low	This trigger is called after a new Event has been saved in the database from a PULL operation. This trigger executes in place of `event-after-save-new`	0	×	✓	2	2024-06-13 11:54:07	☐	×	
Event Before Save	event	high	This trigger is called before an Event or any of its elements is about to be saved in the database		✓	✓				×	
Event Publish	event	low	This trigger is called just before a MISP Event starts the publishing process	50	✓	✓	1	2024-10-07 13:31:02	☐	✓	
Event Report After Save	event-report	low	This trigger is called after an Event Report has been saved in the database	30	×	✓	5	2024-10-04 09:04:54	☐	×	
Log After Save	log	high	This trigger is called after a Log event has been saved in the database	0	×	×	4	2024-07-18 09:02:02	☐	×	
Object After Save	object	high	This trigger is called after an Object has been saved in the database	3	×	✓	12	2024-10-17 08:45:34	☐	×	
Post After Save	post	low	This trigger is called after a Post has been saved in the database		×	×				×	
Shadow Attribute After Save	shadow-attribute	medium	This trigger is called just after a Shadow Attribute has been saved in the database	25	×	✓	7	2024-10-07 13:34:04	☐	×	
Shadow Attribute Before Save	shadow-attribute	medium	This trigger is called just before a Shadow Attribute is saved in the database	0	✓	✓	6	2024-10-04 09:40:03	☐	×	
Sighting After Save	sighting	medium	This trigger is called when a sighting has been saved	0	×	✓	3	2024-06-18 14:51:10	☐	✓	
Tag Attached After Save	tag	high	This trigger is called just after a Tag has been attached to an Event or an Attribute.	3547	×	✓	9	2024-10-18 09:01:11	☐	✓	
User After Save	user	low	This trigger is called after a user has been saved in the database	0	×	×	10	2024-10-07 13:38:26	☐	×	
User Before Save	user	low	This trigger is called just before a user is save in the database		✓	×				×	

Page 1 of 1, showing 1 records out of 17 total, starting on record 1, ending on 17

MISP Workflow



The plan



- We have an `.onion` address
  - We want to get more info about it ➔ **enrichment**

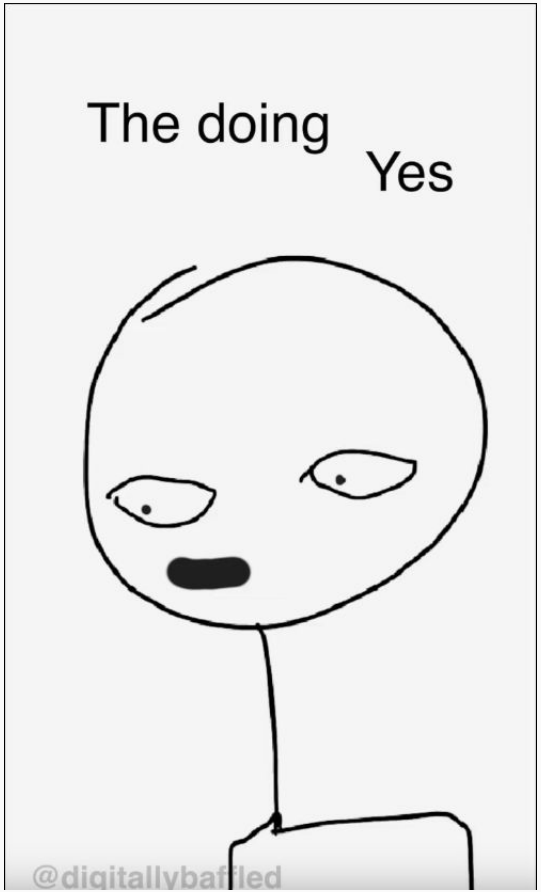
A. **Write the enrichment `onion_lookup.py`**

- We have a way to get more info
  - We want to automate that step ➔ **workflow**

B. **Design a workflow to remove user interaction**

1. Automatically enrich any `.onion` when they enter the tool
2. Restrict sharing if `.onion` contains unwanted content
3. For specific tags warn users on chat application / create a case

A. Write the enrichment `onion_lookup.py`



- STEP 1: Duplicate the pre-made module skeleton

```
$ cp module_misp_standard.py.skeleton \
    onion_lookup.py
```

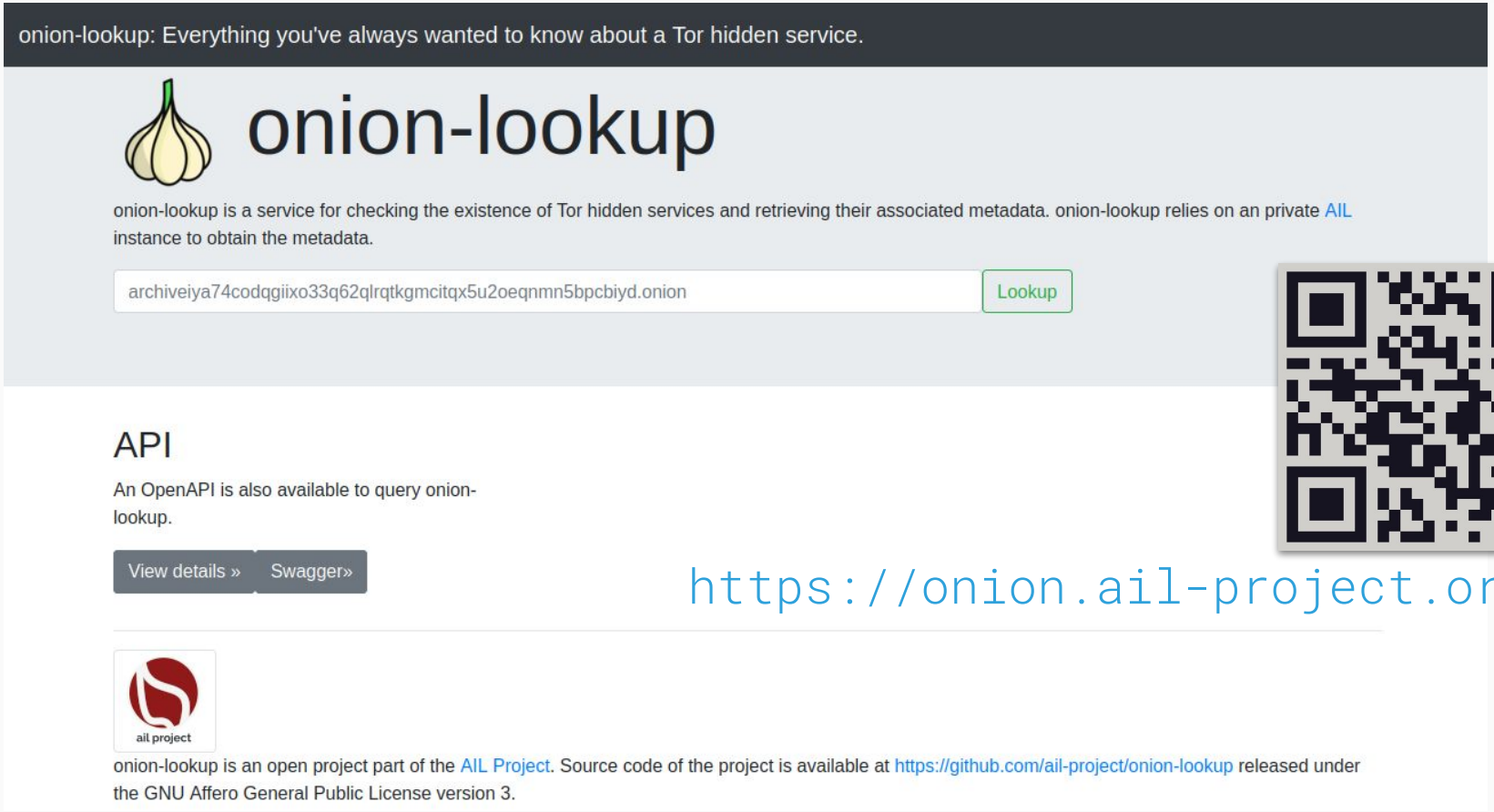
- STEP 2: Modify `onion-lookup.py`

```
$ emacs onion_lookup.py
```

- STEP 3: Restart misp-module

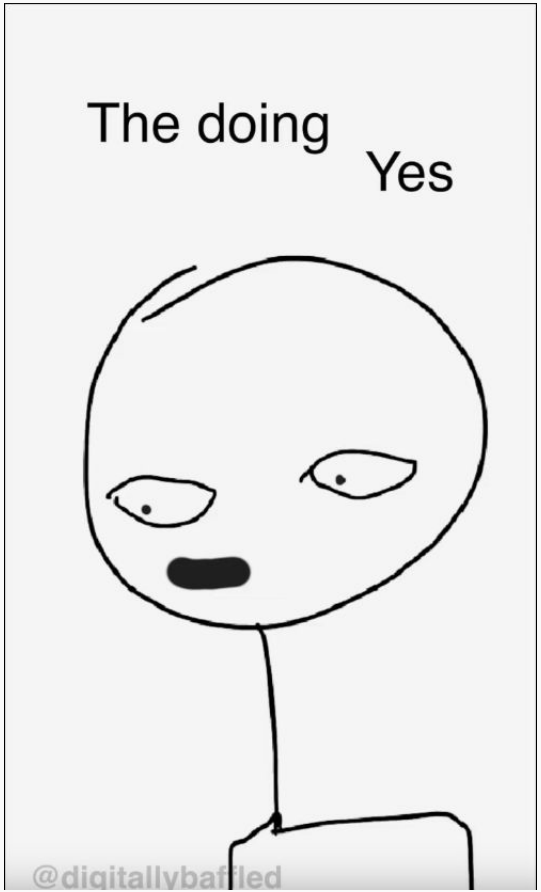
- STEP 4: Test it

Getting information about .onion



https://onion.ail-project.org/

A. Write the enrichment `onion_lookup.py`



- STEP 1: Duplicate the pre-made module skeleton

```
$ cp module_misp_standard.py.skeleton \
    onion_lookup.py
```

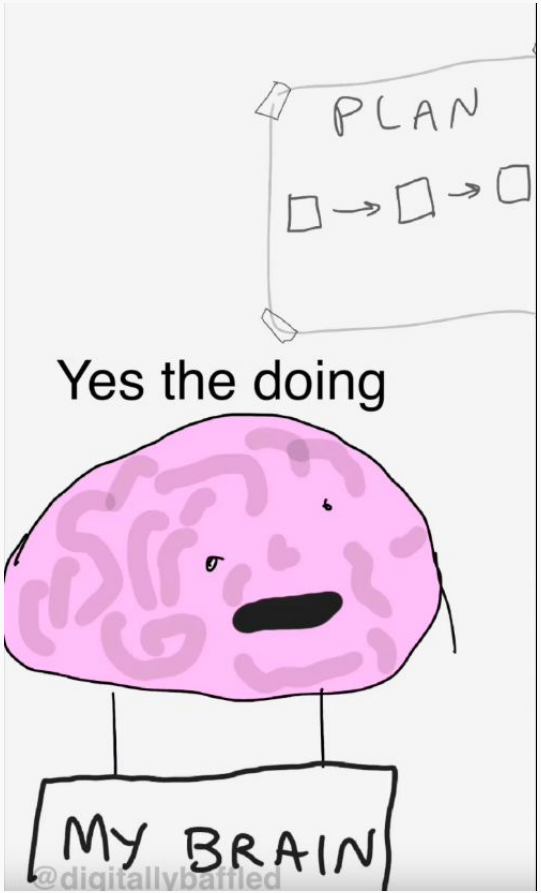
- STEP 2: Modify `onion-lookup.py`

```
$ emacs onion_lookup.py
```

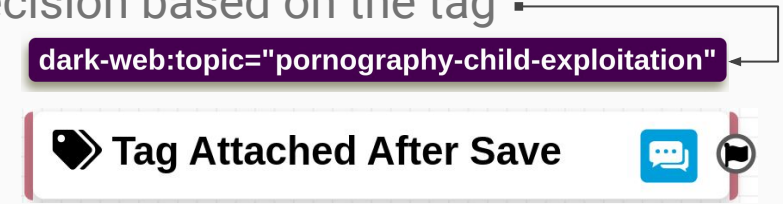
- STEP 3: Restart misp-module

- STEP 4: Test it

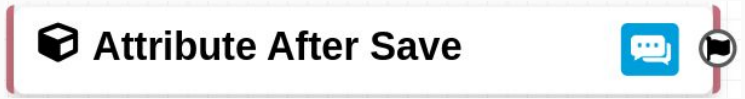
B. Design a workflow to remove user interaction



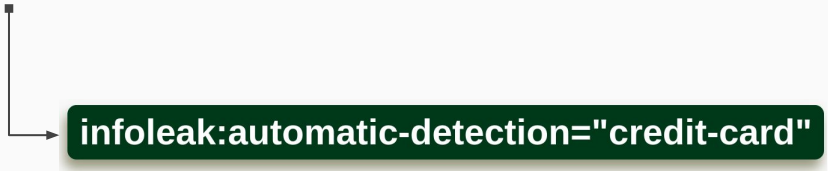
1. Restrict sharing if `.onion` contains unwanted content
   - Decision based on the tag



2. Automatically enrich any `.onion` when they enter the tool



3. For specific tags warn users on chat application / create a case



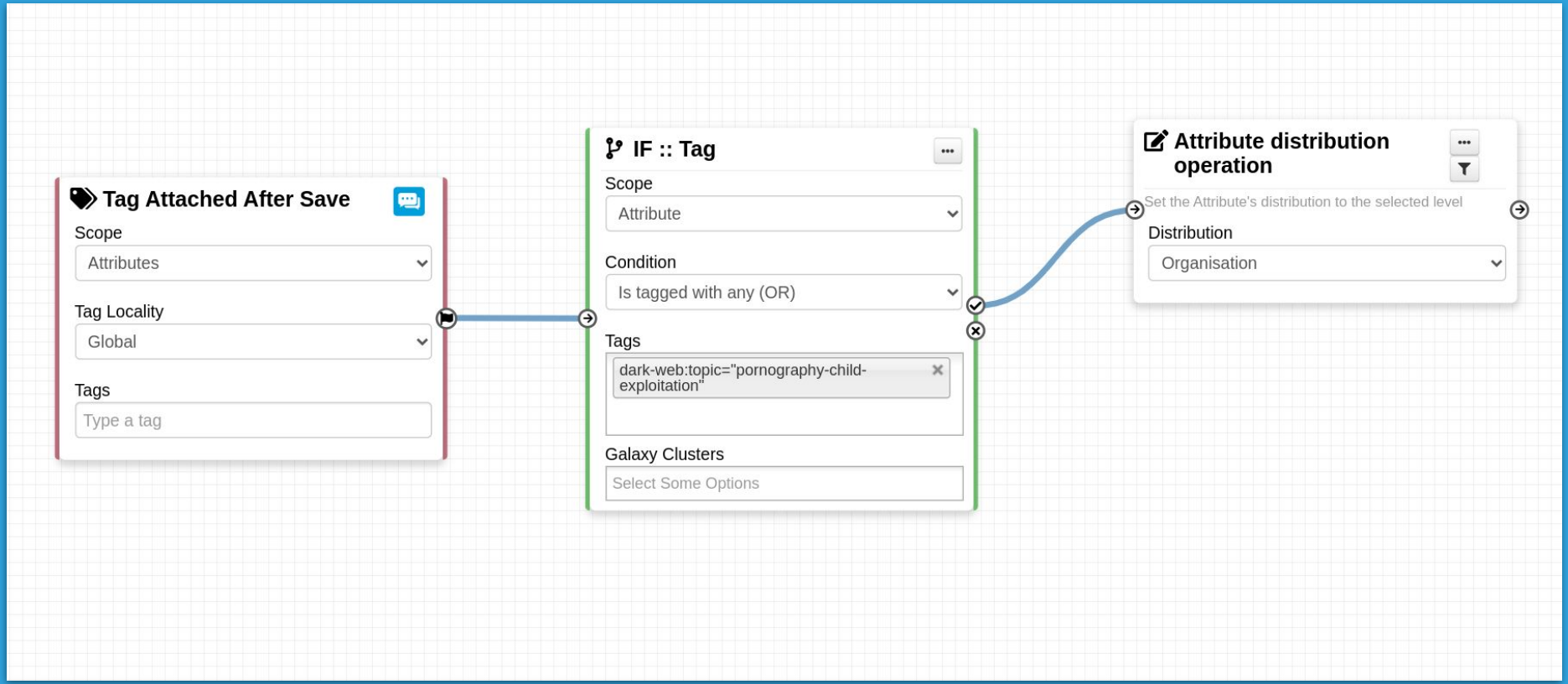
1. Restrict sharing if `.onion` contains unwanted content

Workflow that: **automatically adjusts** attribute's distribution to **Your Org Only**

When `dark-web:topic="pornography-child-exploitation"` is attached.

WHEN tag is attached to attribute
IF tag equals `dark-web:topic="pornography-child-exploitation"`
SET attribute distribution to **Your Org Only**

1. Restrict sharing if `.onion` contains unwanted content



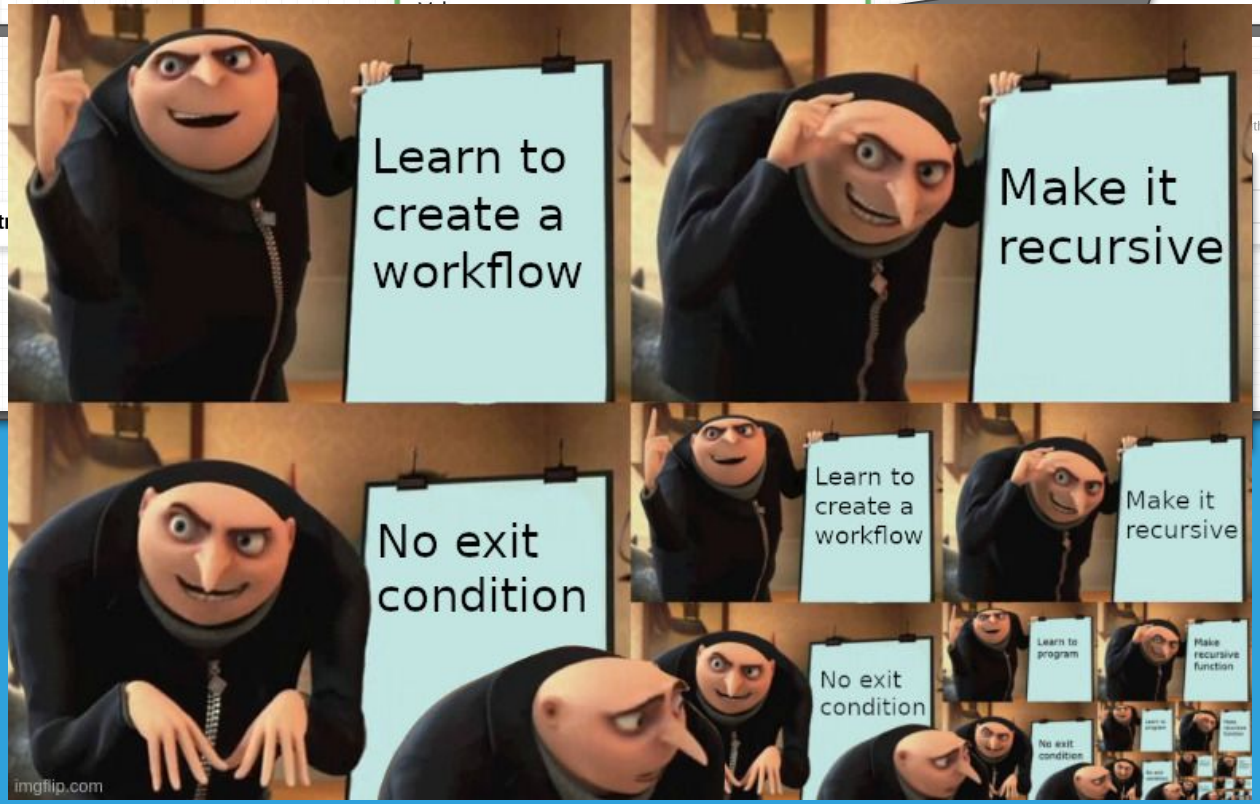
2. Automatically enrich any `.onion` when they enter the tool

Workflow that: **automatically run enrichment** for any `.onion`

WHEN attribute is created

IF attribute is an 

RUN enrichment `onion_lookup.py`

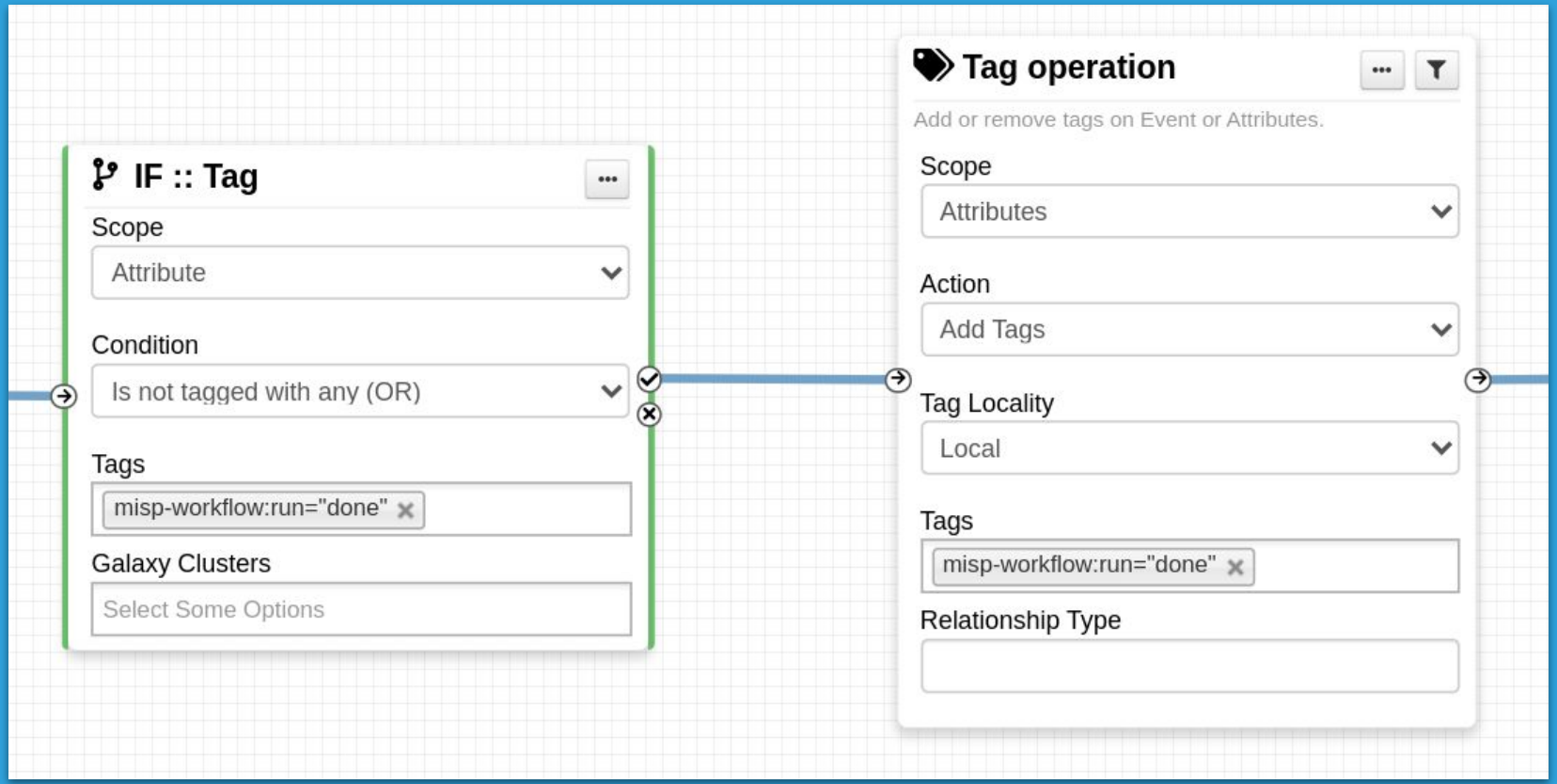

Learn to create a workflow
Make it recursive
No exit condition
Learn to create a workflow
Make it recursive
No exit condition
Learn to program
Make recursive function
No exit condition
imgflip.com

How to solve that recursion?

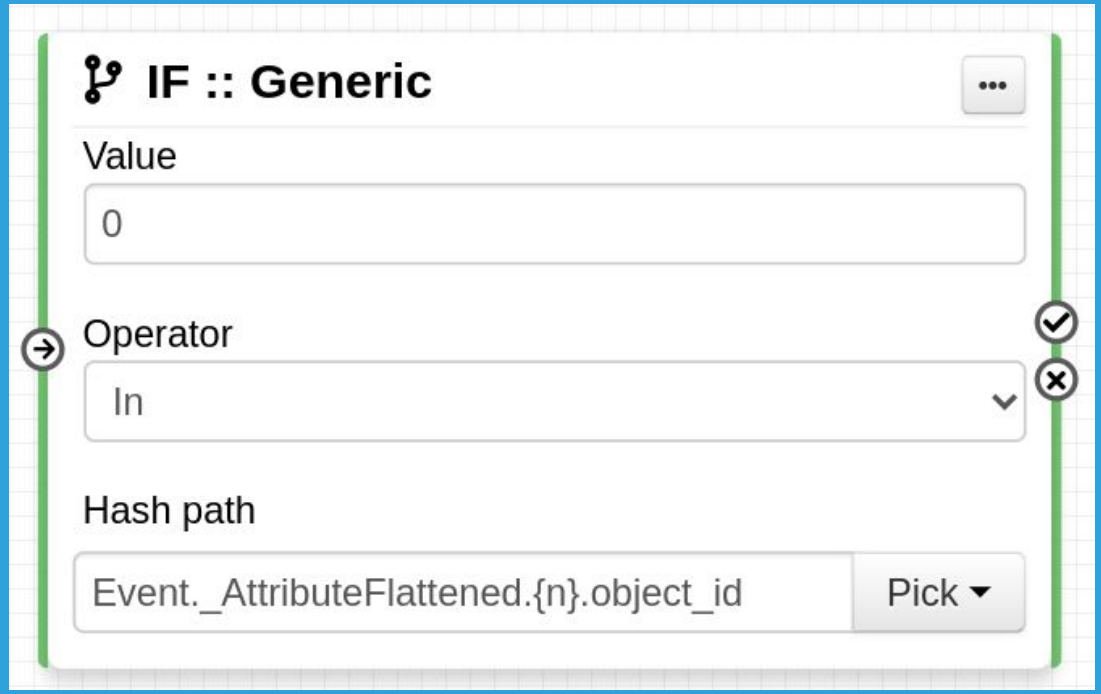
1. Enforce the workflow **to run only once on the same Attribute**
   - Usage of local tags to indicate if it was run or not

2. Prevent the workflow **from running on new Attributes**
   - Only run the workflow on:
     - Attribute, not Object's Attribute
       - Will not work if an analyst creates an `.onion` Object
     - Attribute not having the comment "`created via enrichment`"
       - Requires `onion-lookup.py` to add a comment
       - Requires the workflow to add the comment to new Attributes
     - Attribute not having a dedicated tag
       - Same as above.

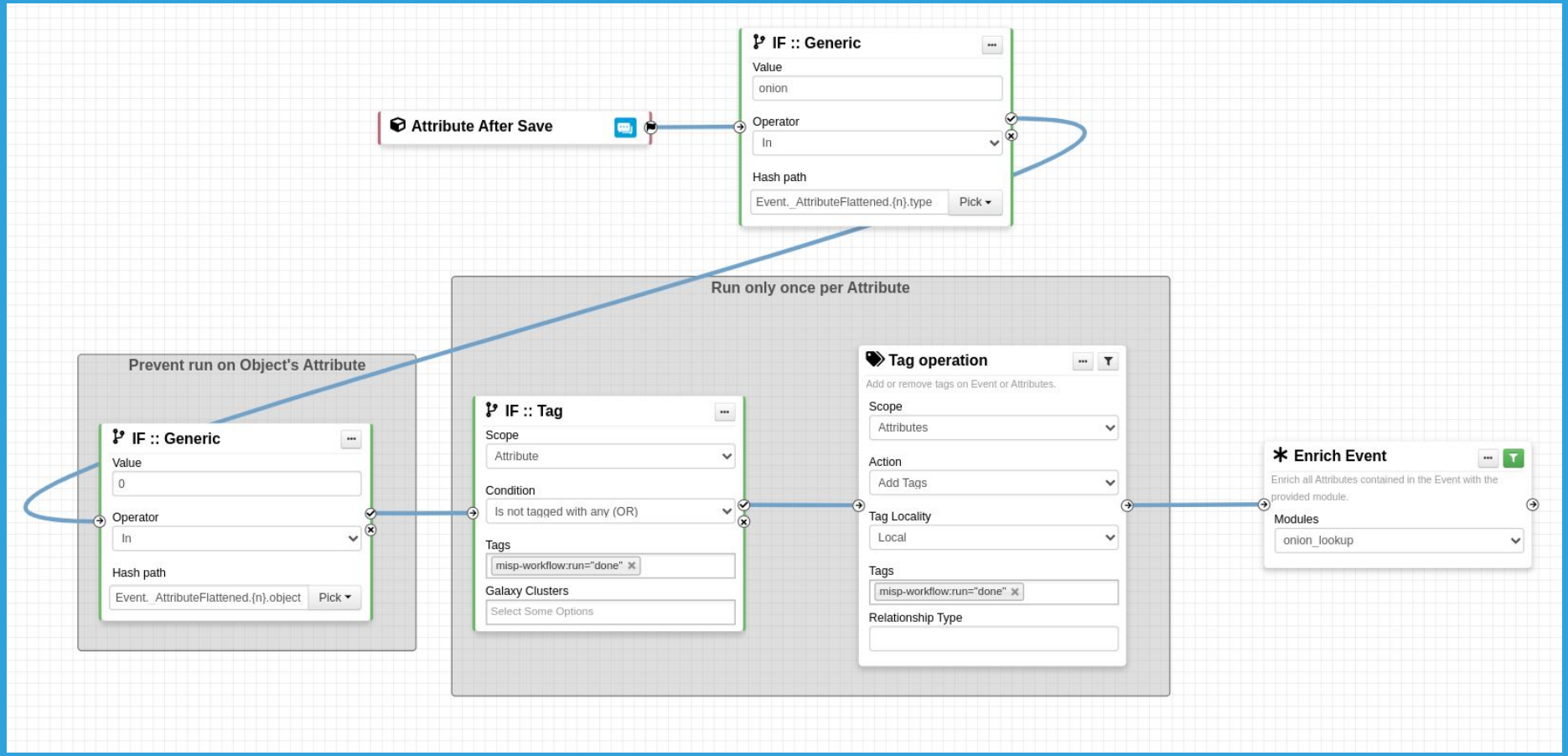
1. Enforce the workflow to run only once on the same Attribute



2. Prevent the workflow from running on new Attributes

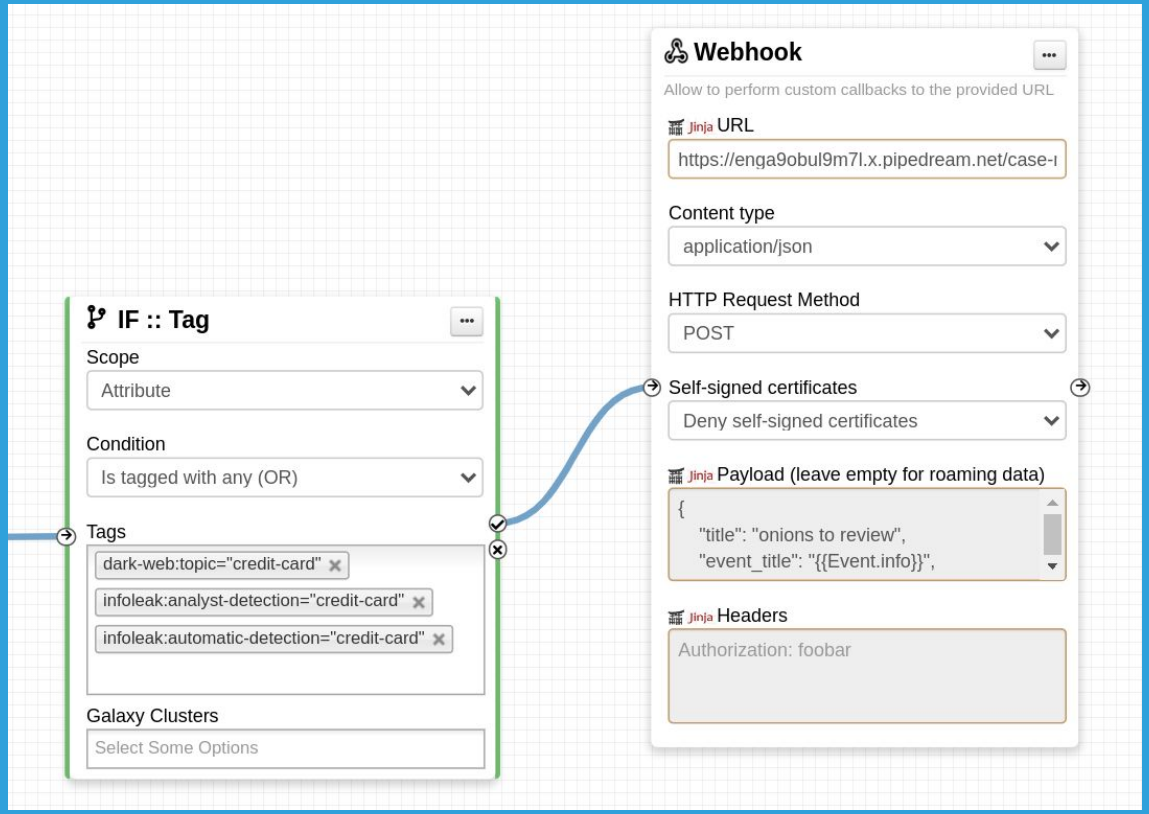


Putting it all together



3. For specific tags warn users on chat application / create a case

infoleak:automatic-detection="credit-card"



The way forward - How better handle recursion

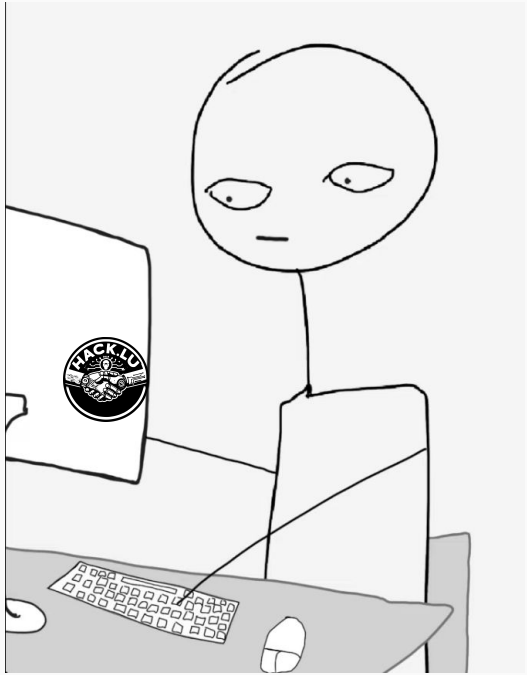
1. Show warning that this workflow might cause a recursion

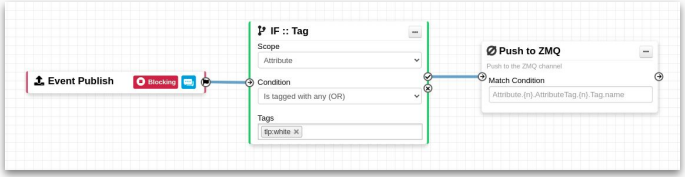
2. Triggerless workflows - Executed manually
   - Manually by an analyst (similar to publishing)
   - *Manually* by another workflow

3. Scheduled workflows - Time-based
   - Need to define how to feed data into the workflow (e.g. newly modified events

Thank you!

misp-module





- All video game icons are sourced from SteamDB or Google Images and are the property of their respective owners.
- Font Awesome Free 6.6.0 by @fontawesome - https://fontawesome.com License - https://fontawesome.com/license/free Copyright 2024 Fonticons, Inc.
- “Doing the plan” from Digitally Baffled - https://www.youtube.com/watch?v=xImh4aGe3ok

Conclusion