MISP

INTELLIGENCE SHARING

Taxonomies and tagging in MISP 2.5

Classify consistently, filter precisely, share deliberately

CIRCL / MISP Project

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

Reference snapshot: 29 October 2026

Make four decisions with every classification

1. Which vocabulary expresses the assessment?
2. Does it describe the event or one attribute?
3. Is it shared context or a local assessment?
4. How will people and tools use it?

By the end

Apply canonical tags, interpret filters and review sharing intent.

Four concepts that answer different questions

Concept	Purpose	Example
Taxonomy	Shared classification vocabulary.	Reliability, confidence, handling.
Tag	A label attached to data.	`tlp:amber`
Galaxy cluster	A described entity or concept with relationships.	A technique or threat actor.
Warninglist	Match values needing extra scrutiny.	Special-use IP ranges.

Read the machine tag, then its explanation

Canonical machine value

```
admiralty-scale:source-reliability="b"
```

Part	Meaning

`admiralty-scale`
`source-reliability`
`"b"`
	Namespace: which vocabulary?
	Predicate: which question?
	Value: the selected answer. Expanded label: Usually reliable.

Some tags have no value: `tlp:amber`.

Separate source reliability from information credibility

What do we know about the source?

`admiralty-scale:source-reliability="b"`
Usually reliable.

What supports this particular information?

`admiralty-scale:information-credibility="2"`
Probably true.

Choose and prepare the vocabulary

Use the TLP 2.0 labels deliberately

Machine tag	Disclosure boundary, in brief
`tlp:clear`	No limit on disclosure, subject to applicable rules.
`tlp:green`	Within the community; no public channels.
`tlp:amber`	Need-to-know within the organisation and its clients.
`tlp:amber+strict`	Need-to-know within the organisation only.
`tlp:red`	Individual recipients only; no further disclosure.

FIRST TLP 2.0: `https://www.first.org/tlp/`

A handling label does not set access controls

Communicate intent

TLP tells recipients the permitted disclosure boundary.

Configure access

Distribution, sharing groups and user permissions govern access in MISP.

Review both before publishing or synchronising.

Prepare a small, useful set of taxonomies

1. Update the taxonomy files in the deployment.
2. Update the taxonomy library in MISP.
3. Enable the chosen taxonomy.
4. Enable all or selected tags from that vocabulary.

Check the result

Find the exact tag in the event or attribute tag picker.

Use governance controls with their actual scope

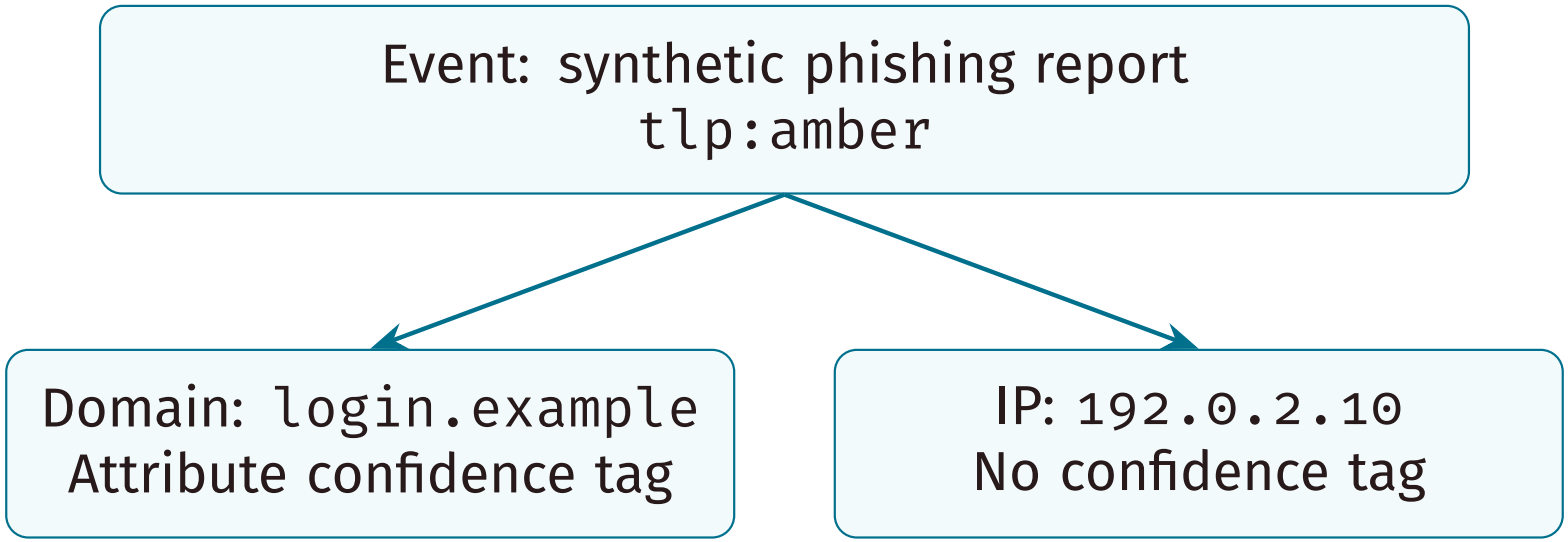
Control	What to expect
Exclusive	A taxonomy or predicate can disallow conflicting choices.
Required	An enabled taxonomy can be required in the UI publish check.
Highlighted	Surface selected taxonomy tags prominently in event views.

For automation

Validate classification explicitly; a UI check is not an API-wide gate.

Apply tags at the right scope

Event context and attribute context are separate



Event tags are not copied onto each stored attribute.

Choose shared context or a local assessment

Shared attachment

Context intended to travel with the data, subject to export and sync rules.

Local attachment

An instance’s assessment; excluded from ordinary external synchronisation.

Local does not mean private to one user.

Three tag settings that are easy to confuse

Setting	Meaning
`local_only`	The tag can only be attached as a local tag.
`exportable`	Whether the tag is eligible for export; other rules still apply.
Organisation / user restriction	Who may apply the tag, not who may read the event.

Inspect tag settings and the attachment's local flag separately.

Use collections to apply a consistent starting set

- A tag collection groups labels for reuse.
- Select a collection from the tag picker.
- Review its meaning and scope before applying it.
- An administrator can configure a default event collection.

A starting point still needs review

A preset cannot know the source’s handling restrictions.

Keep the classification's meaning in context

- Event reports can also carry tags in the reviewed 2.5 source.
- A tag attachment can carry a relationship type.
- Use that annotation to explain how a label relates to its target.
- Keep the canonical taxonomy value unchanged.

An analyst’s tagging sequence

1. Read the evidence and the source’s handling instructions.
2. Choose the event or the specific attribute.
3. Select canonical tags and the intended attachment scope.
4. Review conflicts, distribution and the resulting export.

Worked example: classify a synthetic report

Event

```
tlp:amber
admiralty-scale:source-reliability="b"
```

Domain attribute: `login.example`

```
misp:confidence-level="fairly-confident"
```

IP `192.0.2.10`: no attribute confidence tag.
Event: organisation only; unpublished.

Find, process and review the data

Make filter logic explicit

Operator	Meaning within the selected endpoint and scope
`OR`	Match at least one of these labels.
`AND`	Match every required label.
`NOT`	Exclude data carrying a listed label.

Before interpreting a result

Check the endpoint, attachment scope, permissions and other filters.

Search for attributes with both classifications

`POST /attributes/restSearch`

```
{
  "returnFormat": "json",
  "eventid": "7fb18198-5f8f-4fc3-8ea7-d2d791c2c0be",
  "tags": {
    "AND": [
      "tlp:amber",
      "misp:confidence-level=\"fairly-confident\""
    ]
  },
  "includeEventTags": true
}
```

The domain matches event handling *and* attribute confidence.

A matching attribute can use event context

- Attribute tag searches can also match event tags.
- Request event tags explicitly when you need that context.
- Keep event and attribute provenance clear in consumers.
- Check returned values, not just the number of events.

In the example

`includeEventTags: true` adds context to the response; it does not change stored tags.

Tags help route work; configuration performs it

- Use labels such as `workflow:todo="review"` to coordinate work.
- Configure workflow conditions and actions for automation.
- Configure sync push/pull rules for the intended partner.
- Test a permitted and a rejected case before relying on the rule.

A warninglist hit is a reason to investigate

- Compare an observable against enabled reference lists.
- Consider shared infrastructure, reserved values or common mistakes.
- Read the matching list and why the value appears in it.
- Keep legitimate evidence even when it should not become a detection.

Avoid a false conclusion

A match does not prove that the observable is harmless.

Demonstrate a warning without using a live target

Training value

`192.0.2.10` falls in `192.0.2.0/24`.

- The reviewed RFC 5735 list contains this documentation range.
- Its matching types include `ip-dst`.
- Enable that list before expecting a hit in the exercise.

Choose warning visibility or result suppression

`POST /attributes/restSearch`

Option	Behavior in the reviewed attribute search
`includeWarninglistHits`	Set to `true`: return matches with warning details.
`enforceWarninglist`	Set to `true`: omit matches from enabled applicable lists.

Compare the two supplied request files and inspect the actual response.

Diagnose the cause before creating another tag

Symptom	Check first
Tag missing in picker	Taxonomy, enabled terms, hidden tags and assignment restrictions.
Duplicate-looking labels	Canonical spelling, case and expanded display text.
Label missing downstream	Local/exportable flags, sync route and export options.
Unexpected search results	Endpoint, event versus attribute context and Boolean groups.

Maintain and practise

A taxonomy definition starts with a clear question

```
{
  "namespace": "training-classification",
  "description": "Review state for a local exercise.",
  "version": 1,
  "predicates": [{
    "value": "review",
    "expanded": "Review state",
    "exclusive": true
  }]
}
```

This header is extended by values in the complete example file.

Values connect machine codes to readable labels

```
"values": [{
  "predicate": "review",
  "entry": [
    {"value": "draft", "expanded": "Draft"},
    {"value": "reviewed", "expanded": "Reviewed"}
  ]
}]
```

Resulting machine tag

```
training-classification:review="reviewed"
```

Contribute a vocabulary that others can maintain

1. Establish a gap and cite authoritative definitions.
2. Reuse stable codes and document the intended scope.
3. Validate the schema, predicate links and generated manifest.
4. Submit a focused PR with examples and compatibility notes.

https://github.com/MISP/misp-taxonomies

Use the vocabulary outside a MISP instance

- Read the JSON vocabulary and its canonical values directly.
- Use PyTaxonomies to load and search taxonomy libraries.
- Validate against the current schema and repository checks.
- Record the library revision used by your consumer.

Workshop: classify, search, explain

1. Import the synthetic event into a training instance.
2. Find its event tags and the domain’s confidence tag.
3. Run the scoped tag and warninglist requests.
4. Explain scope, handling, detection intent and every warning.

Then review a local assessment

If permitted, attach a local review label and inspect its local flag.

Before you share or automate

1. Are labels canonical, justified and attached at the right scope?
2. Do distribution and recipients respect handling instructions?
3. Have local context and warninglist hits been reviewed?
4. Does the real consumer interpret returned data correctly?

References and the next practical steps

- Taxonomy reference and FIRST TLP 2.0.
- Taxonomy source and PyTaxonomies.
- Warninglists and MISP 2.5 source.
- This deck's `README.md`: exercise, build instructions, pinned evidence and validation limits.