MISP and Decaying of Indicators

An indicator scoring method and ongoing implementation in MISP

Team CIRCL

info@circl.lu

November 13, 2019
Expiring IOCs: Why and How?
Sharing information about threats is crucial

Organisations are sharing more and more

Contribution by unique organisation (OrgC.name) on MISPPriv:

<table>
<thead>
<tr>
<th>Date</th>
<th>Unique Org</th>
</tr>
</thead>
<tbody>
<tr>
<td>2013</td>
<td>17</td>
</tr>
<tr>
<td>2014</td>
<td>43</td>
</tr>
<tr>
<td>2015</td>
<td>82</td>
</tr>
<tr>
<td>2016</td>
<td>105</td>
</tr>
<tr>
<td>2017</td>
<td>118</td>
</tr>
<tr>
<td>2018</td>
<td>125</td>
</tr>
<tr>
<td>2019-10</td>
<td>135</td>
</tr>
</tbody>
</table>

```json
{
  "distribution": [1, 2, 3]
}
```
Various users and organisations can share data via MISP, multiple parties can be involved

- **Trust, data quality** and **time-to-live** issues
- Each user/organisation has **different use-cases** and interests
  - Conflicting interests such as operational security, attribution,... (depends on the user)

→ Can be partially solved with *Taxonomies*
Various users and organisations can share data via MISP, multiple parties can be involved

- Trust, data quality and time-to-live issues
- Each user/organisation has different use-cases and interests
  - Conflicting interests such as operational security, attribution, ...
    (depends on the user)

→ Can be partially solved with Taxonomies

Attributes can be shared in large quantities (more than 7.3 million on MISPPRIV)

- Partial info about their freshness (Sightings)
- Partial info about their validity (last update)

→ Can be partially solved with our Decaying model
Requirements to enjoy the decaying feature in MISP

- Starting from **MISP 2.4.116**, the decaying feature is available.
- Don’t forget to update the decay models and enable the ones you want.
- The decaying feature has no impact on the information in MISP, it’s just an overlay to be used in the user-interface and API.
- Decay strongly relies on *Taxonomies* and *Sightings*, don’t forget to review their configuration.
**Sightings - Refresher**

*Sightings* add temporal context to indicators. A user, script or an IDS can extend the information related to indicators by reporting back to MISP that an indicator has been seen, or that an indicator can be considered as a false-positive.

- *Sightings* give more credibility/visibility to indicators.
- This information can be used to **prioritise and decay indicators**.
Organisations opt-in - setting a level of confidence

MISP is a peer-to-peer system, information passes through multiple instances.

- Producers can add context (such as tags from Taxonomies, Galaxies) about their asserted confidence or the reliability of the data
- Consumers can have different levels of trust in the producers and/or analysts themselves
- Users might have other contextual needs

→ Achieved thanks to Taxonomies
Tagging is a simple way to attach a classification to an Event or an Attribute

Classification must be globally used to be efficient
### ADMIRALTY-SCALE Taxonomy Library

**Id**: 127  
**Namespace**: admiralty-scale  
**Description**: The Admiralty Scale or Ranking (also called the NATO System) is used to rank the reliability of a source and the credibility of an information. Reference based on FM 2-22.3 (FM 34-52) HUMAN INTELLIGENCE COLLECTOR OPERATIONS and NATO documents.

**Version**: 4  
**Enabled**: Yes (disable)

<table>
<thead>
<tr>
<th>Tag</th>
<th>Expanded</th>
<th>Numerical value</th>
<th>Events</th>
<th>Attributes</th>
<th>Tags</th>
</tr>
</thead>
<tbody>
<tr>
<td>admiralty-scale:information-credibility=&quot;1&quot;</td>
<td>Information Credibility: Confirmed by other sources</td>
<td>100</td>
<td>6</td>
<td>0</td>
<td>admiralty-scale:information-credibility=&quot;1&quot;</td>
</tr>
<tr>
<td>admiralty-scale:information-credibility=&quot;2&quot;</td>
<td>Information Credibility: Probably true</td>
<td>75</td>
<td>21</td>
<td>1</td>
<td>admiralty-scale:information-credibility=&quot;2&quot;</td>
</tr>
<tr>
<td>admiralty-scale:information-credibility=&quot;3&quot;</td>
<td>Information Credibility: Possibly true</td>
<td>50</td>
<td>16</td>
<td>5</td>
<td>admiralty-scale:information-credibility=&quot;3&quot;</td>
</tr>
<tr>
<td>admiralty-scale:information-credibility=&quot;4&quot;</td>
<td>Information Credibility: Doubtful</td>
<td>25</td>
<td>2</td>
<td>0</td>
<td>admiralty-scale:information-credibility=&quot;4&quot;</td>
</tr>
<tr>
<td>admiralty-scale:information-credibility=&quot;5&quot;</td>
<td>Information Credibility: Improbable</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>admiralty-scale:information-credibility=&quot;5&quot;</td>
</tr>
<tr>
<td>admiralty-scale:information-credibility=&quot;6&quot;</td>
<td>Information Credibility: Truth cannot be judged</td>
<td>50</td>
<td>9</td>
<td>2</td>
<td>admiralty-scale:information-credibility=&quot;6&quot;</td>
</tr>
<tr>
<td>admiralty-scale:source-reliability=&quot;a&quot;</td>
<td>Source Reliability: Completely reliable</td>
<td>100</td>
<td>1</td>
<td>0</td>
<td>admiralty-scale:source-reliability=&quot;a&quot;</td>
</tr>
<tr>
<td>admiralty-scale:source-reliability=&quot;b&quot;</td>
<td>Source Reliability: Usually reliable</td>
<td>75</td>
<td>21</td>
<td>76</td>
<td>admiralty-scale:source-reliability=&quot;b&quot;</td>
</tr>
<tr>
<td>admiralty-scale:source-reliability=&quot;c&quot;</td>
<td>Source Reliability: Fairly reliable</td>
<td>50</td>
<td>9</td>
<td>8</td>
<td>admiralty-scale:source-reliability=&quot;c&quot;</td>
</tr>
<tr>
<td>admiralty-scale:source-reliability=&quot;d&quot;</td>
<td>Source Reliability: Not usually reliable</td>
<td>25</td>
<td>2</td>
<td>0</td>
<td>admiralty-scale:source-reliability=&quot;d&quot;</td>
</tr>
<tr>
<td>admiralty-scale:source-reliability=&quot;e&quot;</td>
<td>Source Reliability: Unreliable</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>admiralty-scale:source-reliability=&quot;e&quot;</td>
</tr>
<tr>
<td>admiralty-scale:source-reliability=&quot;f&quot;</td>
<td>Source Reliability: Reliability cannot be judged</td>
<td>50</td>
<td>10</td>
<td>7</td>
<td>admiralty-scale:source-reliability=&quot;f&quot;</td>
</tr>
<tr>
<td>admiralty-scale:source-reliability=&quot;g&quot;</td>
<td>Source Reliability: Deliberately deceptive</td>
<td>0</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
</tr>
</tbody>
</table>

→ Cherry-pick allowed Tags
Some taxonomies have numerical_value
→ Can be used to prioritise Attributes

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Completely reliable</td>
<td>100</td>
</tr>
<tr>
<td>Usually reliable</td>
<td>75</td>
</tr>
<tr>
<td>Fairly reliable</td>
<td>50</td>
</tr>
<tr>
<td>Not usually reliable</td>
<td>25</td>
</tr>
<tr>
<td>Unreliable</td>
<td>0</td>
</tr>
<tr>
<td>Reliability cannot be judged</td>
<td>50?</td>
</tr>
<tr>
<td>Deliberately deceptive</td>
<td>0?</td>
</tr>
</tbody>
</table>
Scoring Indicators: Our solution

\[
score(\text{Attribute}) = base\_score(\text{Attribute, Model}) \cdot decay(\text{Model, time})
\]

Where,

- \( score \in [0, +\infty) \)
- \( base\_score \in [0, 100] \)
- decay is a function defined by model’s parameters controlling decay speed
- Attribute Contains Attribute’s values and metadata (Taxonomies, Galaxies, ...)
- Model Contains the Model’s configuration
Current implementation in MISP
Decay score toggle button

- Shows Score for each Models associated to the Attribute type
Implementation in MISP: API result

/assets/attributes/restSearch

```json
"Attribute": [
{
    "category": "Network activity",
    "type": "ip-src",
    "to_ids": true,
    "timestamp": "1565703507",
    [...]
    "value": "8.8.8.8",
    "decay_score": [
    {
        "score": 54.475223849544456,
        "decayed": false,
        "DecayingModel": {
            "id": "85",
            "name": "NIDS Simple Decaying Model"
        }
    }
    [...]
]}
```
Implementation in MISP: Playing with Models

- **Automatic scoring** based on default values
- **User-friendly UI** to manually set *Model* configuration (lifetime, decay, etc.)
- **Simulation** tool
- Interaction through the **API**
- Opportunity to create your **own** formula or algorithm
Decaying Models in Depth
Scoring Indicators: base_score (1)

\[ \text{score}(\text{Attribute}) = \text{base\_score}(\text{Attribute, Model}) \cdot \text{decay}(\text{Model, time}) \]

When scoring indicators\(^1\), multiple parameters\(^2\) can be taken into account. The base score is calculated with the following in mind:

- Data reliability, credibility, analyst skills, custom prioritisation tags (economical-impact), etc.
- Trust in the source

\[ \text{base\_score} = \omega_{tg} \cdot \text{tags} + \omega_{sc} \cdot \text{source\_confidence} \]

Where,

\[ \omega_{sc} + \omega_{tg} = 1 \]

\(^1\)Paper available: https://arxiv.org/pdf/1803.11052
\(^2\)at a variable extent as required
### Scoring Indicators: base_score (2)

Current implementation ignores source_confidence:

→ base_score = tags

<table>
<thead>
<tr>
<th>Tag</th>
<th>Computation</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>admiralty-scale:source-reliability=&quot; Completely reliable&quot;</td>
<td>Eff. 0.50 * 100.00</td>
<td>50.00</td>
</tr>
<tr>
<td>phishing:psychological-acceptability=&quot;high&quot;</td>
<td>Eff. 0.50 * 75.00</td>
<td>37.50</td>
</tr>
<tr>
<td></td>
<td></td>
<td>87.50</td>
</tr>
</tbody>
</table>

→ The base_score can be used to prioritize attributes based on their attached context and source
Scoring Indicators: decay speed (1)

\[
\text{score}(\text{Attribute}) = \text{base\_score}(\text{Attribute, Model}) \cdot \text{decay}(\text{Model, time})
\]

The decay is calculated using:

- The lifetime of the indicator
  - May vary depending on the indicator type
  - short for an IP, long for an hash
- The decay rate, or speed at which an attribute loses score over time
- The time elapsed since the latest update or sighting
→ decay rate is **re-initialized upon sighting** addition, or said differently, the score is reset to its base score as new *sightings* are applied.

\[
\text{score} = \text{base}\_\text{score} \cdot \left(1 - \left(\frac{t}{\tau}\right)^\frac{1}{\delta}\right)
\]

- $\tau = \text{lifetime}$
- $\delta = \text{decay speed}$
Implementation in MISP: Models definition

\[ \text{score} = \text{base}\_\text{score} \cdot \left(1 - \left(\frac{t}{\tau}\right)^{\frac{1}{\delta}}\right) \]

Models are an instanciation of the formula where elements can be defined:

- Parameters: lifetime, decay_rate, threshold
- base_score
- default base_score
- formula
- associate Attribute types
- creator organisation
Multiple model types are available

- **Default Models**: Models created and shared by the community. Available from misp-decaying-models repository³.
  - → Not editable

- **Organisation Models**: Models created by a user belonging to an organisation
  - These models can be hidden or shared to other organisation
  - → Editable

³https://github.com/MISP/misp-decaying-models.git
## Implementation in MISP: Index

### Decaying Models

<table>
<thead>
<tr>
<th>ID</th>
<th>Organization</th>
<th>Usable to everyone</th>
<th>Name</th>
<th>Description</th>
<th>Parameters {}</th>
<th>Formula</th>
<th># Assigned Types</th>
<th>Version</th>
<th>Enabled</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>29</td>
<td>1</td>
<td>✔</td>
<td>Phishing model</td>
<td>Simple model to rapidly decay phishing website.</td>
<td>{&quot;lifetime&quot;: 3, &quot;decay_speed&quot;: 2.3, &quot;threshold&quot;: 30, &quot;default_base_score&quot;: 80, &quot;base_score_config&quot;: {&quot;estimative-language&quot;: 0.5, &quot;phishing&quot;: 0.5}}</td>
<td>Polynomial</td>
<td>9</td>
<td>1</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>85</td>
<td>1</td>
<td>✗</td>
<td>NIDS Simple Decaying Model</td>
<td>Simple decaying model for Network Intrusion Detection System (NIDS).</td>
<td>{&quot;lifetime&quot;: 120, &quot;decay_speed&quot;: 2, &quot;threshold&quot;: 30, &quot;default_base_score&quot;: 80, &quot;base_score_config&quot;: {&quot;estimative-language&quot;: 0.25, &quot;priority-level&quot;: 0.25, &quot;retention&quot;: 0.25, &quot;targeted-threat-index&quot;: 0.125, &quot;false-positive&quot;: 0.125}}</td>
<td>Polynomial</td>
<td>13</td>
<td>1</td>
<td>✔</td>
<td></td>
</tr>
</tbody>
</table>

Page 1 of 1, showing 2 records out of 2 total, starting on record 1, ending on 2

View, update, add, create, delete, enable, export, import
Implementation in MISP: Fine tuning tool

Create, modify, visualise, perform mapping
**IMPLEMENTATION IN MISP: base_score TOOL**

<table>
<thead>
<tr>
<th>Search Taxonomy</th>
<th>3 not having numerical value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default base score</td>
<td>80</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Taxonomies</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Admiralty-scale</td>
<td></td>
</tr>
<tr>
<td>Source-reliability</td>
<td>31</td>
</tr>
<tr>
<td>Information-credibility</td>
<td>30</td>
</tr>
<tr>
<td>Priority-level</td>
<td></td>
</tr>
<tr>
<td>Priority-level</td>
<td>53</td>
</tr>
<tr>
<td>Retention</td>
<td>0</td>
</tr>
<tr>
<td>Estimative-language</td>
<td></td>
</tr>
<tr>
<td>Likelihood-probability</td>
<td>0</td>
</tr>
<tr>
<td>Confidence-in-analytic-judgment</td>
<td>0</td>
</tr>
<tr>
<td>Misp</td>
<td></td>
</tr>
<tr>
<td>Confidence-level</td>
<td>0</td>
</tr>
<tr>
<td>Threat-level</td>
<td>0</td>
</tr>
<tr>
<td>Automation-level</td>
<td>0</td>
</tr>
<tr>
<td>Phishing</td>
<td></td>
</tr>
<tr>
<td>State</td>
<td>0</td>
</tr>
<tr>
<td>Psychological-acceptability</td>
<td>0</td>
</tr>
</tbody>
</table>

**Example**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Tags</th>
<th>Base score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tag your attribute</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Attribute 1</td>
<td>Admiralty-scale:information-credibility=&quot;5&quot;</td>
<td>0.0</td>
</tr>
<tr>
<td>Attribute 2</td>
<td>Priority-level:baseline-minor, Admiralty-scale:source-reliability=&quot;d&quot;, Admiralty-scale:information-credibility=&quot;2&quot;</td>
<td>38.2</td>
</tr>
<tr>
<td>Attribute 3</td>
<td>Priority-level:severe, Admiralty-scale:information-credibility=&quot;2&quot;</td>
<td>84.6</td>
</tr>
</tbody>
</table>

**Computation steps**

<table>
<thead>
<tr>
<th>Tag</th>
<th>Computation</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority-level:baseline-minor</td>
<td>Eff. Ratio, Value</td>
<td>11.62</td>
</tr>
<tr>
<td>Admiralty-scale:source-reliability=&quot;d&quot;</td>
<td>Eff. Ratio, Value</td>
<td>0.80</td>
</tr>
</tbody>
</table>
Simulate Attributes with different Models
/attributes/restSearch

```json
{
    "includeDecayScore": 1,
    "includeFullModel": 0,
    "excludeDecayed": 0,
    "decayingModel": [85],
    "modelOverrides": {
        "threshold": 30
    }
    "score": 30,
}
```
Creating a new decay algorithm (1)

The current architecture allows users to create their own formulae.

1. Create a new file $filename in 
   app/Model/DecayingModelsFormulas/
2. Extend the Base class as defined in DecayingModelBase
3. Implement the two mandatory functions computeScore and isDecayed using your own formula/algorithm
4. Create a Model and set the formula field to $filename

Use cases:

- Add support for **more feature** (expiration taxonomy)
- **Query external services** then influence the score
- Completely **different approach** (i.e streaming algorithm)
- ...
<?php
include_once 'Base.php';

class Polynomial extends DecayingModelBase
{
    public const DESCRIPTION = 'The description of your new decaying algorithm';

    public function computeScore($model, $attribute, $base_score, $elapsed_time)
    {
        // algorithm returning a numerical score
    }

    public function isDecayed($model, $attribute, $score)
    {
        // algorithm returning a boolean stating if the attribute is expired or not
    }
}
Decaying Models 2.0

- Improved support of Sightings
  - False positive Sightings should somehow reduce the score
  - Expiration Sightings should mark the attribute as decayed

- Potential Model improvements
  - Instead of resetting the score to base_score once a Sighting is set, the score should be increased additively (based on a defined coefficient); thus prioritizing surges rather than infrequent Sightings
  - Take into account related Tags or Correlations when computing score

- Increase Taxonomy coverage
  - Users should be able to manually override the numerical_value of Tags

- For specific type, take into account data from other services
  - Could fetch data from BGP ranking, Virus Total, Passive X for IP/domain/... and adapt the score