# MISP and Decaying of Indicators

## Primer for indicator scoring in MISP

Team CIRCL

info@circl.lu

September 13, 2022




MISP and Decaying of Indicators
Primer for indicator scoring in MISP
Team CIRCL
info@circl.lu
September 13, 2022
MISP
Threat Sharing

# Outline of the presentation

- Present the components used in MISP to expire IOCs
- Present the current state of Indicators life-cycle management in MISP





Outline of the presentation
- Present the components used in MISP to expire IOCs
- Present the current state of Indicators life-cycle management in MISP

# Expiring IOCs: Why and How?





Expiring IOCs: Why and How?

# Indicators lifecycle - Problem Statement

- **Sharing information** about threats **is crucial**
- Organisations are sharing more and more

Contribution by **unique organisation** (`Orgc.name`) on MISPPriv:

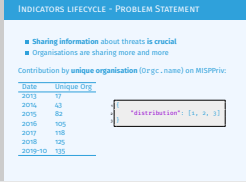
| Date | Unique Org |
|---|---|
| 2013 | 17 |
| 2014 | 43 |
| 2015 | 82 |
| 2016 | 105 |
| 2017 | 118 |
| 2018 | 125 |
| 2019-10 | 135 |

```
{
    "distribution": [1, 2, 3]
}
```









# Indicators lifecycle - Problem Statement

- Various users and organisations can share data via MISP, multiple parties can be involved
  - **Trust**, **data quality** and **relevance** issues
  - Each user/organisation have **different use-cases** and interests
    - Conflicting interests: Operational security VS attribution

→ Can be partially solved with *Taxonomies*





Indicators lifecycle - Problem Statement
- Various users and organisations can share data via MISP, multiple parties can be involved
  - Trust, data quality and relevance issues
  - Each user/organisation have different use-cases and interests
    - Conflicting interests: Operational security VS attribution

→ Can be partially solved with Taxonomies

# Indicators lifecycle - Problem Statement

- Various users and organisations can share data via MISP, multiple parties can be involved
  - **Trust**, **data quality** and **relevance** issues
  - Each user/organisation have **different use-cases** and interests
    - Conflicting interests: Operational security VS attribution

  → Can be partially solved with *Taxonomies*

- Attributes can be shared in large quantities (more than 12M on MISPPRIV - Sept. 2020)
  - Partial info about their **freshness** (*Sightings*)
  - Partial info about their **validity** (*last_seen*)

  → Can be partially solved with our *Data model*

  MISP's *Decaying model* combines the two





Indicators lifecycle - Problem Statement
Various users and organisations can share data via MISP, multiple parties can be involved
Trust, data quality and relevance issues
Each user/organisation have different use-cases and interests
Conflicting interests: Operational security VS attribution
→ Can be partially solved with Taxonomies
Attributes can be shared in large quantities (more than 12M on MISPPRIV - Sept. 2020)
Partial info about their freshness (Sightings)
Partial info about their validity (last_seen)
→ Can be partially solved with our Data model
MISP's Decaying model combines the two

# Requirements to enjoy the decaying feature in MISP

- Starting from **MISP 2.4.116**, the decaying feature is available
- **Update** decay models and **enable** some
- MISP Decaying strongly relies on *Taxonomies* and *Sightings*, don't forget to review their configuration

Note: The decaying feature has no impact on the information stored in MISP, it's just an **overlay** to be used in the user-interface and API







Requirements to enjoy the decaying feature in MISP
Starting from MISP 2.4.116, the decaying feature is available
Update decay models and enable some
MISP Decaying strongly relies on Taxonomies and Sightings, don't forget to review their configuration
Note: The decaying feature has no impact on the information stored in MISP, it's just an overlay to be used in the user-interface and API

# *Sightings* - Refresher (1)

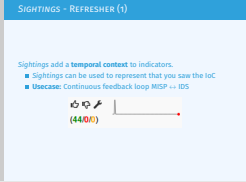
*Sightings* add a **temporal context** to indicators.

- *Sightings* can be used to represent that you saw the IoC
- **Usecase:** Continuous feedback loop MISP ↔ IDS







# *Sightings* - Refresher (2)

*Sightings* add a **temporal context** to indicators.

- *Sightings* give more credibility/visibility to indicators
- This information can be used to **prioritise and decay indicators**







*Sightings* - Refresher (2)
*Sightings* add a **temporal context** to indicators.
- *Sightings* give more credibility/visibility to indicators
- This information can be used to **prioritise and decay indicators**

# Taxonomies - Refresher (1)

**Taxonomies**

« previous | 1 | 2 | next »

| Id ↑ | Namespace | Description | Version | Enabled | Required | Active Tags | Actions |
|---|---|---|---|---|---|---|---|
| 181 | workflow | Workflow support language is a common language to support intelligence analysts to perform their analysis on data and information. | 9 | Yes | | **27** / 26 (enable all) | |
| 180 | vocabulaire-des-probabilites-estimatives | Ce vocabulaire attribue des valeurs en pourcentage à certains énoncés de probabilité | 2 | Yes | | **5** / 5 | |
| 179 | threats-to-dns | An overview of some of the known attacks related to DNS as described by Torabi, S., Boukhtouta, A., Assi, C., & Debbabi, M. (2018) in Detecting Internet Abuse by Analyzing Passive DNS Traffic: A Survey of Implemented Systems. IEEE Communications Surveys & Tutorials, 1–1. doi:10.1109/comst.2018.2849614 | 1 | No | | **0** / 18 | |
| 178 | targeted-threat-index | The Targeted Threat Index is a metric for assigning an overall threat ranking score to email messages that deliver malware to a victim's computer. The TTI metric was first introduced at SecTor 2013 by Seth Hardy as part of the talk "RATastrophe: Monitoring a Malware Menagerie" along with Katie Kleemola and Greg Wiseman. | 2 | Yes | | **11** / 11 | |

- *Taxonomies* are a simple way to attach a classification to an *Event* or an *Attribute*
- Classification must be globally used to be efficient (or agreed on beforehand)







Taxonomies - Refresher (1)
Taxonomies
- Taxonomies are a simple way to attach a classification to an Event or an Attribute
- Classification must be globally used to be efficient (or agreed on beforehand)

# Taxonomies - Refresher (2)

## 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) |

« previous next »

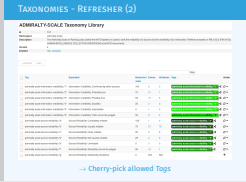
Filter

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

→ Cherry-pick allowed *Tags*







# Taxonomies - Refresher (3)

- Some taxonomies have a `numerical_value`
- Allows concepts to be used in an mathematical expression
  → Can be used to prioritise IoCs

`admirality-scale` taxonomy[1]

| Description | Value |
|---|---|
| Completely reliable | 100 |
| Usually reliable | 75 |
| Fairly reliable | 50 |
| Not usually reliable | 25 |
| Unreliable | 0 |
| Reliability cannot be judged | 50 |
| Deliberatly deceptive | 0 |

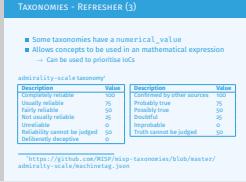
| Description | Value |
|---|---|
| Confirmed by other sources | 100 |
| Probably true | 75 |
| Possibly true | 50 |
| Doubtful | 25 |
| Improbable | 0 |
| Truth cannot be judged | 50 |

[1] `https://github.com/MISP/misp-taxonomies/blob/master/admiralty-scale/machinetag.json`









# Taxonomies - Refresher (3)

admiralty-scale taxonomy[2]

| Description | Value |
|---|---|
| Completely reliable | 100 |
| Usually reliable | 75 |
| Fairly reliable | 50 |
| Not usually reliable | 25 |
| Unreliable | 0 |
| Reliability cannot be judged | 50 ? |
| Deliberatly deceptive | 0 ? |

| Description | Value |
|---|---|
| Confirmed by other sources | 100 |
| Probably true | 75 |
| Possibly true | 50 |
| Doubtful | 25 |
| Improbable | 0 |
| Truth cannot be judged | 50 ? |

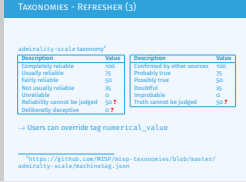
→ Users can override tag numerical_value

[2] https://github.com/MISP/misp-taxonomies/blob/master/admiralty-scale/machinetag.json









# Scoring Indicators: Our solution

$$\texttt{score}(\texttt{Attribute}) = \texttt{base\_score}(\texttt{Attribute}, \texttt{Model}) \bullet \texttt{decay}(\texttt{Model}, \texttt{time})$$

- base_score(Attribute, Model)
  - Initial score of the *Attribute* only considering the context (*Attribute's type*, *Tags*)

- decay(Model, time)
  - Function composed of the **lifetime** and **decay speed**
  - Decreases the base_score over time



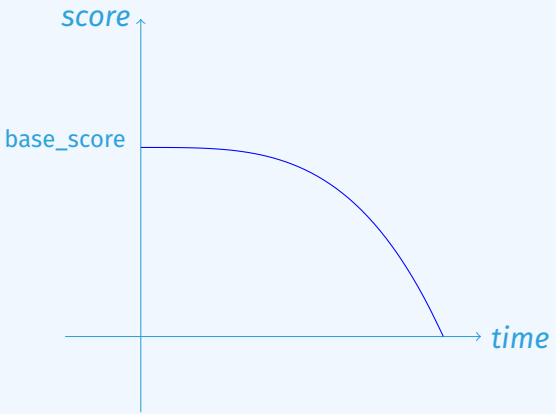




Scoring Indicators: Our solution
score(Attribute) = base_score(Attribute, Model) • decay(Model, time)
base_score(Attribute, Model)
Initial score of the Attribute only considering the context (Attribute's type, Tags)
decay(Model, time)
Function composed of the lifetime and decay speed
Decreases the base_score over time

# Scoring Indicators: Our solution

$$\texttt{score}(\texttt{Attribute}) = \texttt{base\_score}(\texttt{Attribute, Model}) \bullet \texttt{decay}(\texttt{Model, time})$$





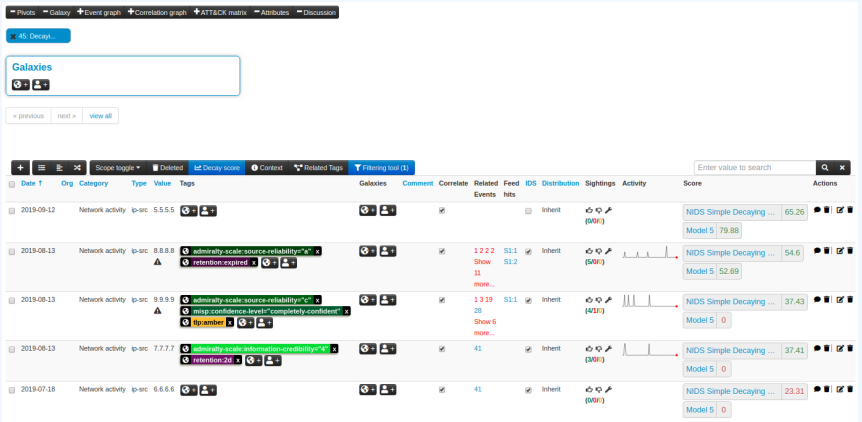





Scoring Indicators: Our solution

# Current implementation in MISP

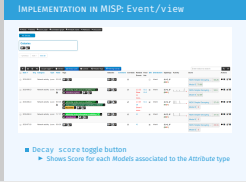
# Implementation in MISP: Event/view




- Decay score toggle button
  - Shows Score for each *Models* associated to the *Attribute* type







# Implementation in MISP: API result

/attributes/restSearch

```
"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: API result
/attributes/restSearch
"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: Objectives

- **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





Implementation in MISP: Objectives
- **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

# Implementation in MISP: Models definition

$$\overset{\curvearrowright}{score} = base\_score \cdot \left(1 - \left(\frac{t}{\tau}\right)^{\frac{1}{\delta}}\right)$$

*Models* are an instanciation of the formula with configurable parameters:

- Parameters: `lifetime, decay_rate, threshold`
- `base_score` computation
- `default base_score`
- associate *Attribute* types
- formula
- creator organisation





Implementation in MISP: Models definition
Models are an instanciation of the formula with configurable parameters:
Parameters: lifetime, decay_rate, threshold
base_score computation
default base_score
associate Attribute types
formula
creator organisation

# Implementation in MISP: Models Types

Two types of model are available

- **Default Models**: Created and shared by the community. Coming from `misp-decaying-models` repository[3].
  → Not editable

- **Organisation Models**: Created by a user on MISP
  - Can be hidden or shared to other organisation
  → Editable

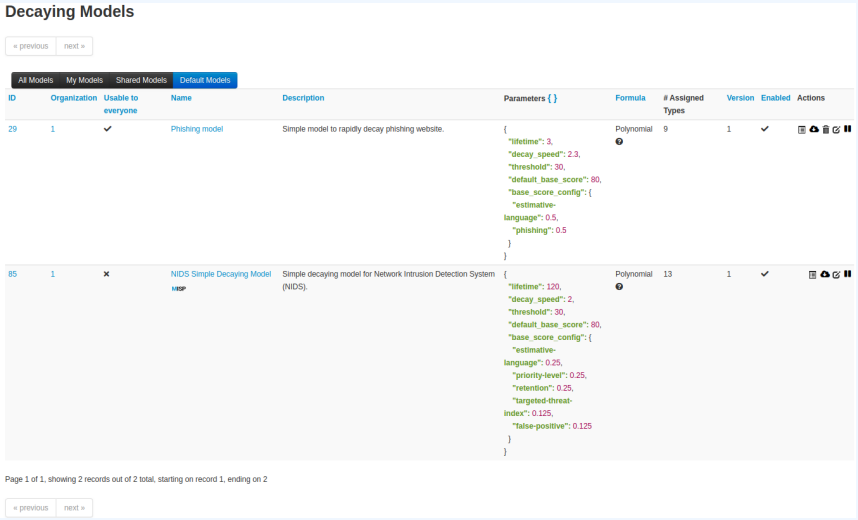
[3] https://github.com/MISP/misp-decaying-models.git





Implementation in MISP: Models Types
Two types of model are available
Default Models: Created and shared by the community. Coming from misp-decaying-models repository[3].
→ Not editable
Organisation Models: Created by a user on MISP
Can be hidden or shared to other organisation
→ Editable
[3] https://github.com/MISP/misp-decaying-models.git

# Implementation in MISP: Index



**Decaying Models**

« previous | next »

All Models | My Models | Shared Models | Default Models

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

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

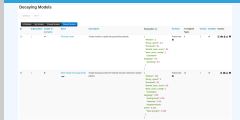
« previous | next »

Standard CRUD operations: View, update, add, create, delete, enable, export, import



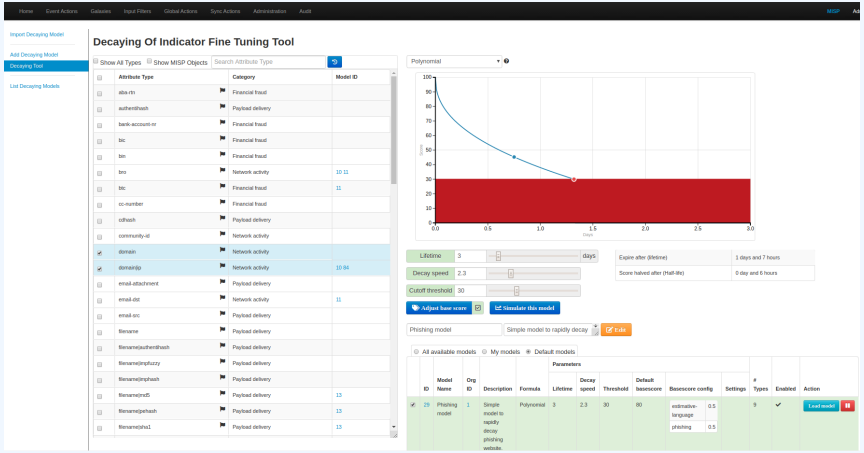


Implementation in MISP: Index



Standard CRUD operations: View, update, add, create, delete, enable, export, import

# Implementation in MISP: Fine tuning tool




Configure models: Create, modify, visualise, perform mapping

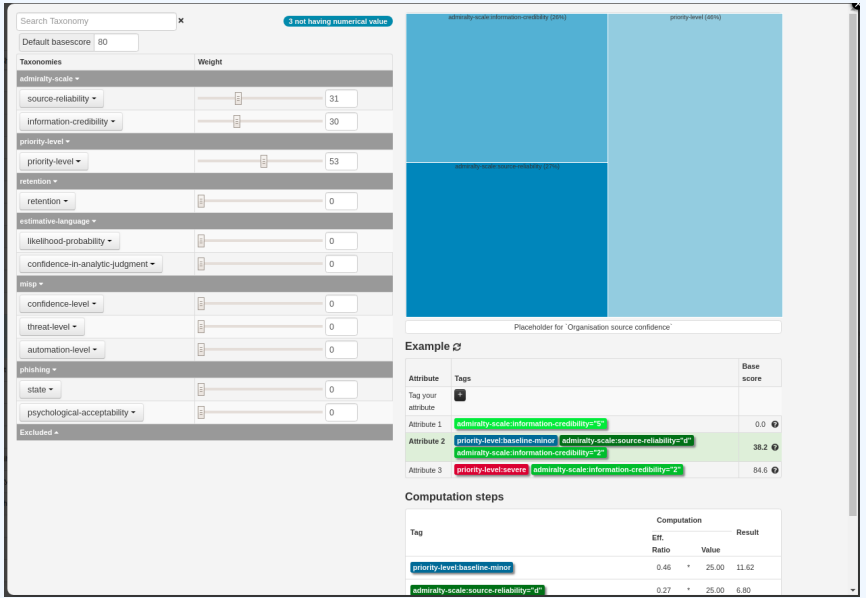




Implementation in MISP: Fine tuning tool

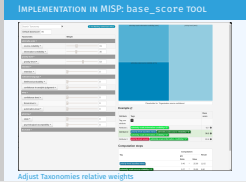
Configure models: Create, modify, visualise, perform mapping

# Implementation in MISP: base_score tool

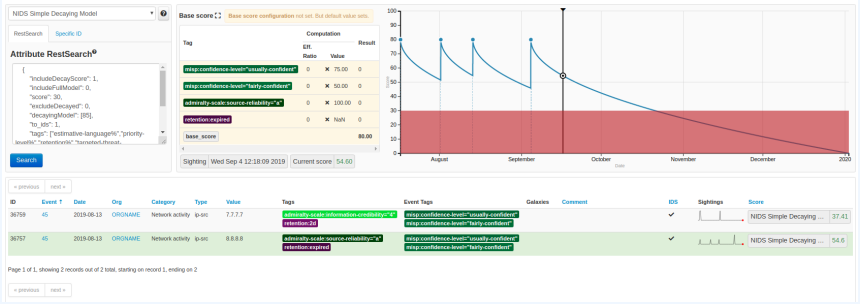






# Implementation in MISP: simulation tool




Simulate decay on *Attributes* with different *Models*





Implementation in MISP: simulation tool

Simulate decay on Attributes with different Models

# Implementation in MISP: API query body

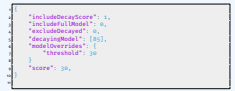
/attributes/restSearch

```
{
    "includeDecayScore": 1,
    "includeFullModel": 0,
    "excludeDecayed": 0,
    "decayingModel": [85],
    "modelOverrides": {
        "threshold": 30
    }
    "score": 30,
}

```





Implementation in MISP: API query body
/attributes/restSearch

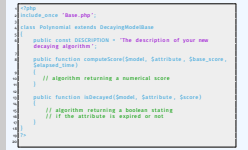


# Creating a new decay algorithm

```php
<?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*





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