MISP

INTELLIGENCE SHARING

Turning data into actionable intelligence

advanced features in MISP supporting your analysts and tools

CIRCL / Team MISP Project

MISP Project

CRI MISP 2024

about CIRCL

circl.lu
Computer Incident
Response Center
LUXEMBOURG

- The Computer Incident Response Center Luxembourg (CIRCL) is a government-driven initiative designed to provide a systematic response facility to computer security threats and incidents. CIRCL is the CERT for the private sector, communes and non-governmental entities in Luxembourg and is operated by securitymadein.lu g.i.e.

MISP and CIRCL

- CIRCL is mandated by the Ministry of Economy and acting as the Luxembourg National CERT for private sector.
- CIRCL leads the development of the Open Source MISP threat intelligence platform which is used by many military or intelligence communities, private companies, financial sector, National CERTs and LEAs globally.
- **CIRCL runs multiple large MISP communities performing active daily threat-intelligence sharing**.

The aim of this presentation

- To give some insight into what sort of an evolution of our various communities' have gone through as observed over the past 8 years
- Show the importance of **strong contextualisation**...
- ...and how that can be leveraged when trying to make our data **actionable**

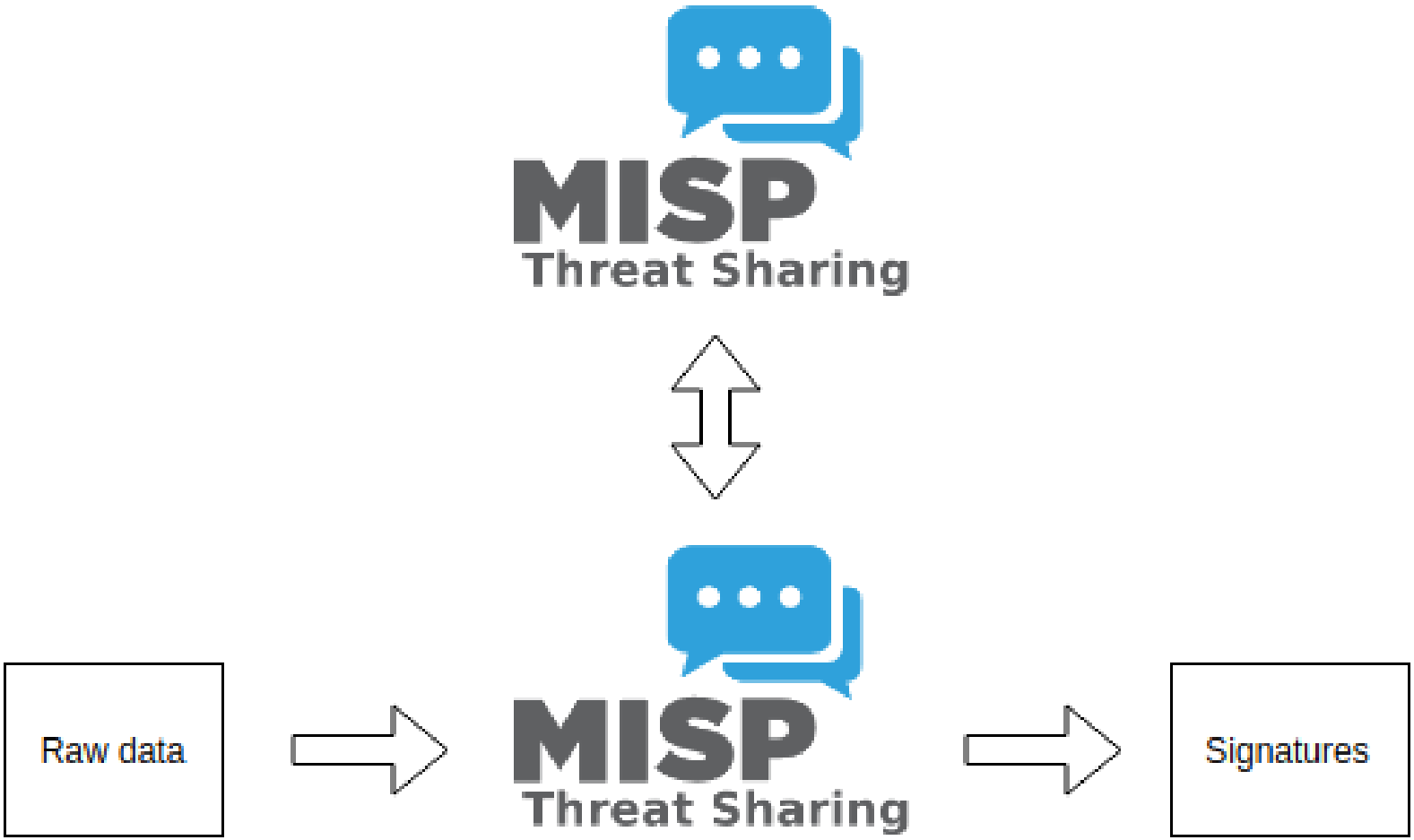
Development based on practical user feedback

- There are many different types of users of an information sharing platform like MISP:
  - **Malware reversers** willing to share indicators of analysis with respective colleagues.
  - **Security analysts** searching, validating and using indicators in operational security.
  - **Intelligence analysts** gathering information about specific adversary groups.
  - **Law-enforcement** relying on indicators to support or bootstrap their DFIR cases.
  - **Risk analysis teams** willing to know about the new threats, likelyhood and occurences.
  - **Fraud analysts** willing to share financial indicators to detect financial frauds.

The initial scope of MISP

- **Extract information** during the analysis process
- Store and **correlate** these datapoints
- **Share** the data with partners
- Focus on technical indicators: IP, domain, hostname, hashes, filename, pattern in file/memory/traffic
- Generate protective signatures out of the data: snort, suricata, OpenIOC

Initial workflow



Why was it so simplistic?

- This was both a reflection of our maturity as a community
  - Capabilities for **extracting** information
  - Capabilities for **utilising** the information
  - Lack of **willingness** to share context
  - Lack of **co-operation** between teams doing technical analysis/monitoring and threat-intel
- The more growth we saw in maturity, the more we tried to match it with our data-model, often against pushback

The growing need to contextualise data

- There were separate factors that made our data-sets less and less useful for detection/defense in general
  - **Growth of our communities**
  - Distinguish between information of interest and raw data
  - **False-positive** management
  - TTPs and aggregate information may be prevalent compared to raw data (risk assessment)
  - **Increased data volumes** leads to be able to prioritise

Our initial solution

- Allow users to **tag any information** created in MISP
- We wanted to be **lax with what we accept** in terms of data, but be **strict on what we fed to our tools**, with strong filter options
- We had some ideas on how to potentially move forward...

Our initial failures

- Try to capture different aspects of contextualisation into **normalised values** (threat level, source reliability, etc)
  - Didn't scale with needs other than our own
  - Incorporating new types of contextualisation would mean **the modification of the software**
  - Getting communities with **established naming conventions** to use anything but their go-to vocabularies was a pipe-dream
  - Heated arguments over numeric conversions

Human creativity

- We tried an alternate approach instead: Free tagging
  - Result was spectacularly painful, at least 7 different ways to spell tlp:amber
  - No canonisation for common terms lead to tagging ultimately becoming a highly flawed tool for filtering within a sharing community

How we ended up tackling the issue more successfuly

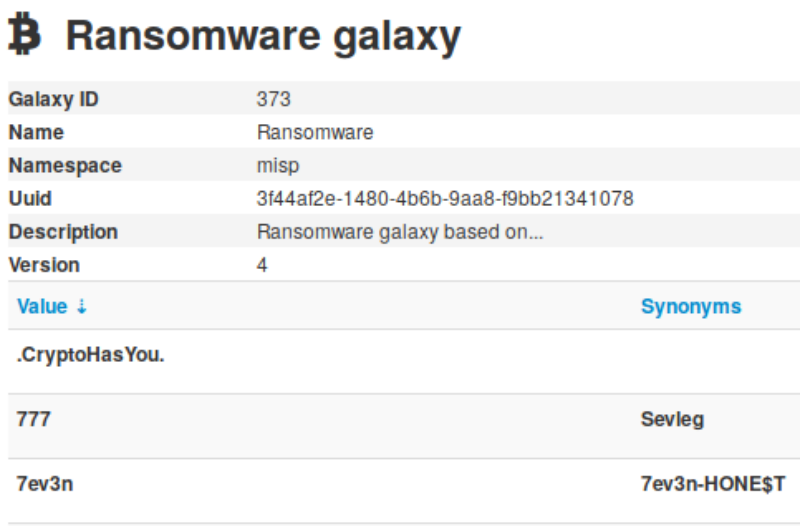
- We ended up with a mixed approach, currently implemented by the MISP-taxonomy system
  - Taxonomies are **vocabularies** of known tags
  - Tags would be in a **triple tag format**

    `namespace:predicate="value"`
  - Create your own taxonomies, recipients should be able to use data you tag with them without knowing it at the first place
  - Avoid any coding, stick to **JSON**
- Massive success, approaching 100 taxonomies
- Organisations can solve their own issues without having to rely on us

Tag	Events	Attributes	Tags
workflow:state="complete"	11	0	workflow:state="complete"
workflow:state="draft"	0	0	workflow:state="draft"
workflow:state="incomplete"	55	10	workflow:state="incomplete"
workflow:state="ongoing"	0	0	workflow:state="ongoing"

We were still missing something...

- Taxonomy tags often **non self-explanatory**
- Example: universal understanding of tlp:green vs APT 28
- For the latter, a single string was ill-suited
- So we needed something new in addition to taxonomies - **Galaxies**
  - Community driven **knowledge-base libraries used as tags**
  - Including descriptions, links, synonyms, meta information, etc.
  - Goal was to keep it **simple and make it reusable**
  - Internally it works the exact same way as taxonomies (stick to **JSON**)

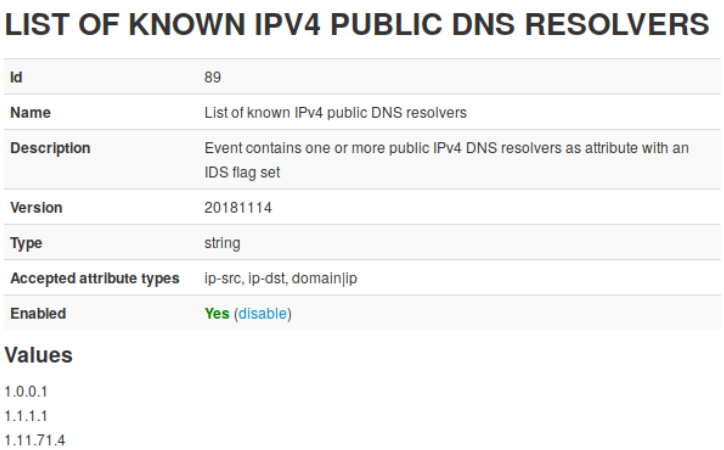


Broadening the scope of what sort of context we are interested in

- **Who** can receive our data? **What** can they do with it?
- **Data accuracy, source reliability**
- **Why** is this data relevant to us?
- **Who** do we think is behind it, **what tools** were used?
- What sort of **motivations** are we dealing with? Who are the **targets**?
- How can we **block/detect/remediate** the attack?
- What sort of **impact** are we dealing with?

Parallel to the contextualisation efforts: False positive handling

- Low quality / false positive prone information being shared
- Lead to **alert-fatigue**
- Exclude organisation xy out of the community?
- False positives are often obvious - **can be encoded**
- **Warninglist system**[1] aims to do that
- Lists of well-known indicators which are often false-positives like RFC1918 networks, ...



LIST OF KNOWN IPV4 PUBLIC DNS RESOLVERS

Id	89
Name	List of known IPv4 public DNS resolvers
Description	Event contains one or more public IPv4 DNS resolvers as attribute with an IDS flag set
Version	20181114
Type	string
Accepted attribute types	ip-src, ip-dst, domain|ip
Enabled	Yes (disable)

Values

1.0.0.1
1.1.1.1
1.11.71.4

Warning: Potential false positives

List of known IPv4 public DNS resolvers
Top 1000 website from Alexa
List of known google domains

[1] https://github.com/MISP/misp-warninglists

More complex data-structures for a modern age

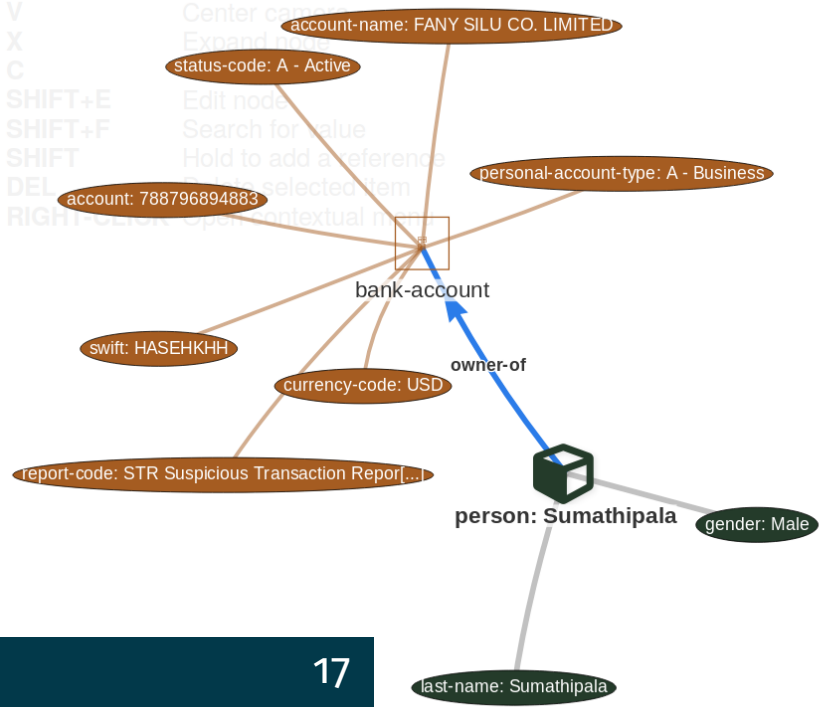
- Atomic attributes were a great starting point, but lacking in many aspects
- **MISP objects**[2] system
  - Simple **templating** approach
  - Use templating to build more complex structures
  - Decouple it from the core, allow users to **define their own** structures
  - MISP should understand the data without knowing the templates
  - Massive caveat: **Building blocks have to be MISP attribute types**
  - Allow **relationships** to be built between objects

[2] https://github.com/MISP/misp-objects

Supporting specific datamodel

Filters: All | File | Network | Financial | Proposal | Correlation | Warnings | Include deleted attributes | Show context fields

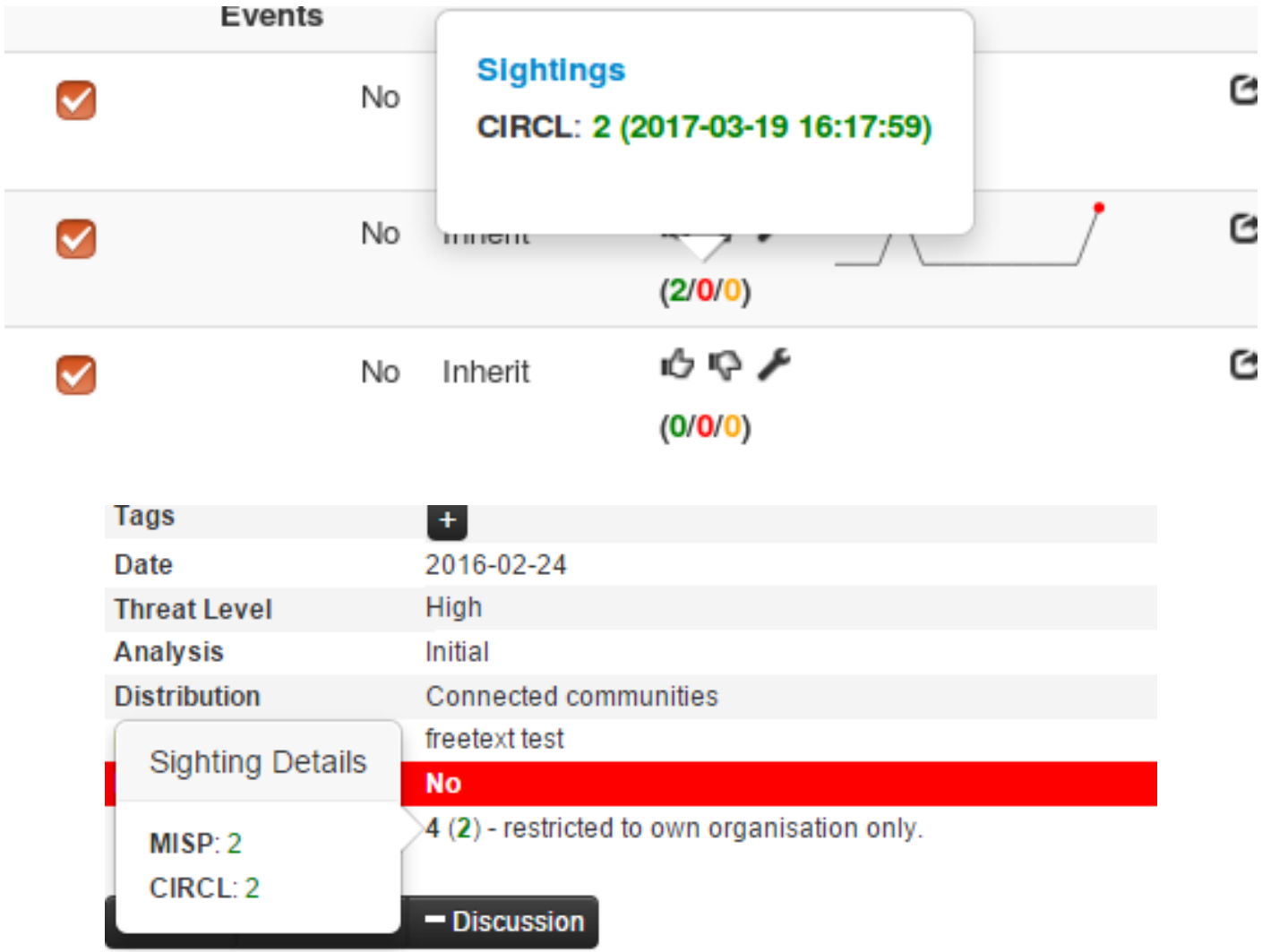
Date	Org	Category	Type	Value	Tags	Galaxies	Comment	Correlate	Related Events
2018-09-28		**Name:** bank-account **References:** 0							
2018-09-28		Other	**status-code:** text	A - Active	+	Add		☐	
2018-09-28		Other	**report-code:** text	STR Suspicious Transaction Report	+	Add		☐	
2018-09-28		Other	**personal-account-type:** text	A - Business	+	Add		☐	
2018-09-28		Financial fraud	**swift:** bic	HASEHKHH	+	Add		☑	3849 11320 11584
2018-09-28		Financial fraud	**account:** bank-account-nr	788796894883	+	Add		☑	
2018-09-28		Other	**account-name:** text	FANY SILU CO. LIMITED	+	Add		☑	
2018-09-28		Other	**currency-code:** text	USD	+	Add		☐	



Continuous feedback loop

- Data ingested by MISP was in a sense frozen in time
- We had a creation data, but lacked a way to use the output of our detection
- Lead to the introduction of the **Sighting system**
- The community could sight indicators and convey the time of sighting
- Potentially powerful tool for IoC lifecycle management, clumsy query implementation default

Supporting specific datamodel



Making use of all this context

- Most obvious goal: Improve the way we query data
  - Unified all export APIs
  - Incorporate all contextualisation options into **API filters**
  - Allow for an **on-demand** way of **excluding potential false positives**
  - Allow users to easily **build their own** export modules feed their various tools

Example query

/attributes/restSearch

```
{
    "returnFormat": "netfilter",
    "enforceWarninglist": 1,
    "tags": {
      "NOT": [
        "tlp:white",
        "type:OSINT"
      ],
      "OR": [
        "misp-galaxy:threat-actor=\"Sofacy\"",
        "misp-galaxy:sector=\"Chemical\""
      ],
    }
}
```

Synchronisation filters

- Make decisions on whom to share data with based on context
  - MISP by default decides based on the information creator's decision who data gets shared with
  - Community hosts should be able to **act as a safety net** for sharing
    - **Push filters** - what can I push?
    - **Pull filters** - what am I interested in?
    - **Local tags** allow for information flow control

The emergence of ATT&CK and similar galaxies

- Standardising on high-level **TTPs** was a solution to a long list of issues
- Adoption was rapid, tools producing ATT&CK data, familiar interface for users
- A much better take on kill-chain phases in general
- Feeds into our **filtering** and **situational awareness** needs extremely well
- Gave rise to other, ATT&CK-like systems tackling other concerns
  - **attck4fraud** [3] by Francesco Bigarella from ING
  - **Election guidelines** [4] by NIS Cooperation Group

[3] https://www.misp-project.org/galaxy.html#_attck4fraud

[4] https://www.misp-project.org/galaxy.html#_election_guidelines

Example query to generate ATT&CK heatmaps

```
/events/restSearch
{
    "returnFormat": "attack",
    "tags": [
        "misp-galaxy:sector=\"Chemical\""
    ],
    "timestamp": "365d"
}
```

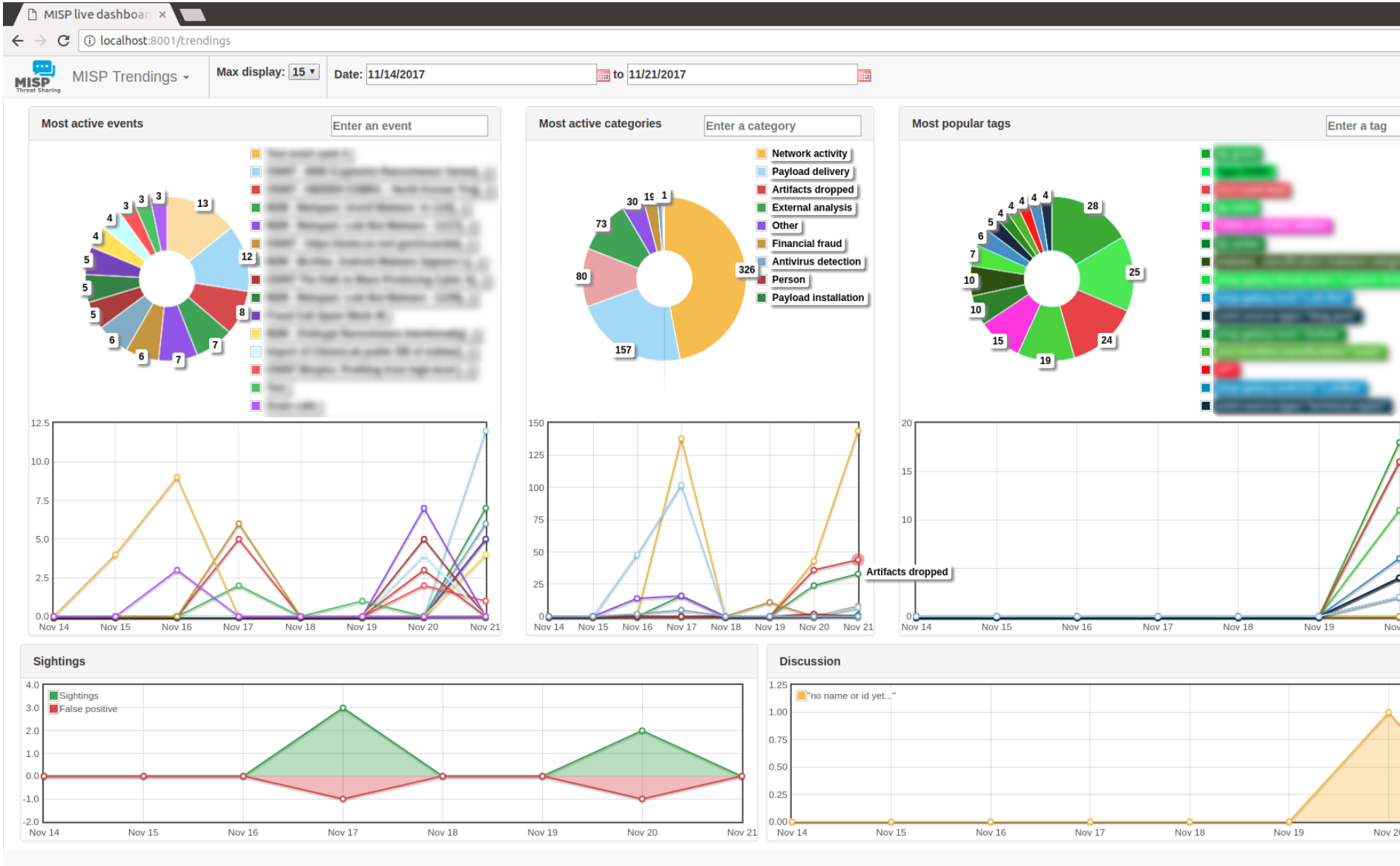
A sample result for the above query

Pre Attack - Attack Pattern | **Enterprise Attack - Attack Pattern** | Mobile Attack - Attack Pattern

0 — 11 — Show all

Initial access	Execution	Persistence	Privilege escalation	Defense evasion	Credential access	Discovery	Lateral movement	Collection	Exfiltration	Command and control
Spearphishing Attachment	Scripting	Screensaver	File System Permissions Weakness	Process Hollowing	Securityd Memory	Password Policy Discovery	AppleScript	Data from Information Repositories	Exfiltration Over Alternative Protocol	Standard Application Layer Protocol
Spearphishing via Service	Command-Line Interface	Login Item	AppCert DLLs	Code Signing	Input Capture	System Network Configuration Discovery	Distributed Component Object Model	Data from Removable Media	Exfiltration Over Command and Control Channel	Communication Through Removable Media
Trusted Relationship	User Execution	Trap	Application Shimming	Rootkit	Bash History	Process Discovery	Pass the Hash	Man in the Browser	Data Compressed	Custom Command and Control Protocol
Replication Through Removable Media	Regsvcs/Regasm	System Firmware	Scheduled Task	NTFS File Attributes	Exploitation for Credential Access	Network Share Discovery	Exploitation of Remote Services	Data Staged	Automated Exfiltration	Multi-Stage Channels
Exploit Public-Facing Application	Trusted Developer Utilities	Registry Run Keys / Start Folder	Startup Items	Exploitation for Defense Evasion	Private Keys	Peripheral Device Discovery	Remote Desktop Protocol	Screen Capture	Scheduled Transfer	Remote Access Tools
Spearphishing Link	Windows Management Instrumentation	LC_LOAD_DYLIB Addition	New Service	Network Share Connection Removal	Brute Force	Account Discovery	Pass the Ticket	Email Collection	Data Encrypted	Uncommonly Used Port
Valid Accounts	Service Execution	LSASS Driver	Sudo Caching	Process Doppelgänging	Password Filter DLL	System Information Discovery	Windows Remote Management	Clipboard Data	Exfiltration Over Other Network Medium	Multilayer Encryption
Supply Chain Compromise	CMSTP	Rc.common	Process Injection	Disabling Security Tools	Two-Factor Authentication Interception	System Network Connections Discovery	Windows Admin Shares	Video Capture	Exfiltration Over Physical Medium	Domain Fronting
Drive-by Compromise	Control Panel Items	Authentication Package	Bypass User Account Control	Timestomp	LLMNR/NBT-NS Poisoning	Network Service Scanning	Remote Services	Audio Capture	Data Transfer Size Limits	Data Obfuscation
Hardware Additions	Dynamic Data Exchange	Component Firmware	Extra Window Memory Injection	Modify Registry	Credentials in Files	File and Directory Discovery	Taint Shared Content	Data from Network Shared Drive		Connection Proxy
	Source	Windows Management Instrumentation Event Subscription	Setuid and Setgid	Indicator Removal from Tools	Forced Authentication	Security Software Discovery	Application Deployment Software	Data from Local System		Commonly Used Port
	Space after Filename	Change Default File	Launch Daemon	Hidden Window	Keychain	System Service Discovery	Third-party Software	Automated Collection		Data Encoding

Monitor trends outside of MISP (example: dashboard)



Decaying of indicators

- We were still missing a way to use all of these systems in combination to decay indicators
- Move the decision making **from complex filter options to** complex **decay models**
- Decay models would take into account various **taxonomies**, **sightings**, the **type** of each indicator **Sightings** and **Creation date**
- The first iteration of what we have in MISP now took:
  - 2 years of research
  - 3 published research papers
  - A lot of prototyping

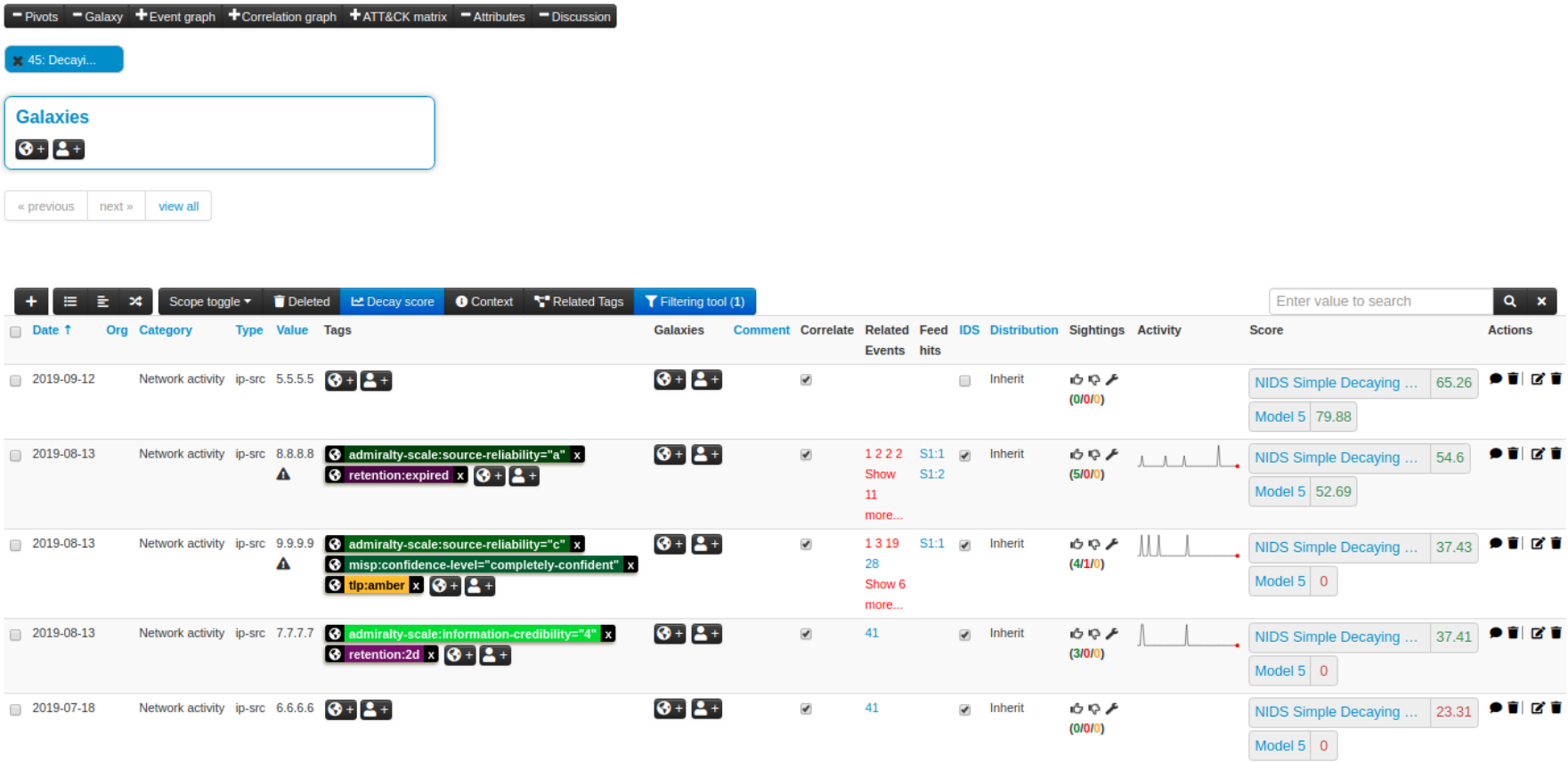
Scoring Indicators: Our solution

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

Where,

- $\texttt{score} \in [0, 100]$
- $\texttt{base_score} \in [0, 100]$
- `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

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": 54.4752238495444456,
        "decayed": false,
        "DecayingModel": {
          "id": "85",
          "name": "NIDS Simple Decaying Model"
        }
```

Implementation in MISP: Index

Decaying Models

« previous | next »

All Models | My Models | Shared Models | Default Models

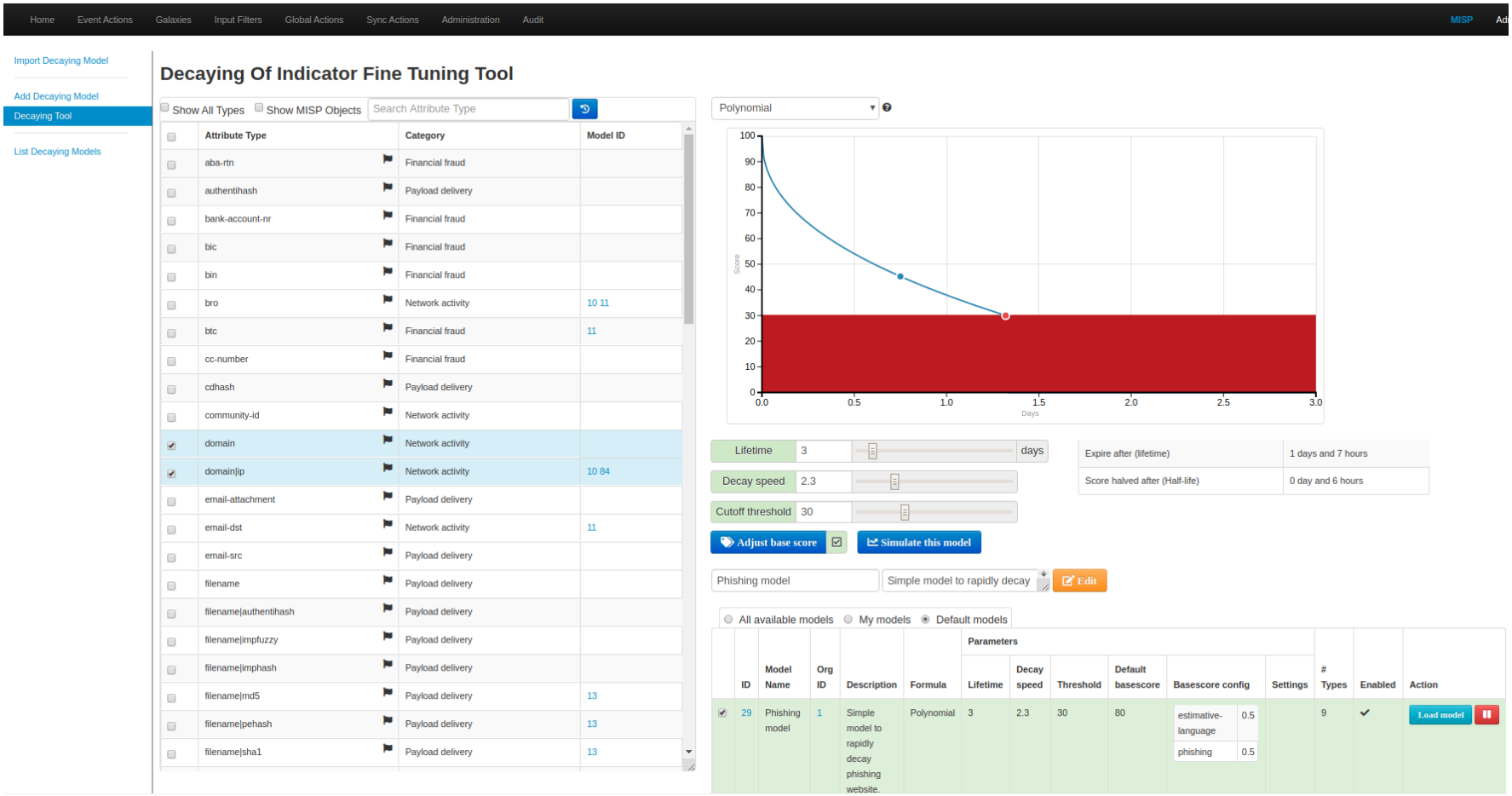
ID	Organization	Usable to everyone	Name	Description	Parameters { }	Formula	# Assigned Types	Version	Enabled	Actions
29	1	✔	Phishing model	Simple model to rapidly decay phishing website.	{ "lifetime": 3, "decay_speed": 2.3, "threshold": 30, "default_base_score": 80, "base_score_config": { "estimative-language": 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 »

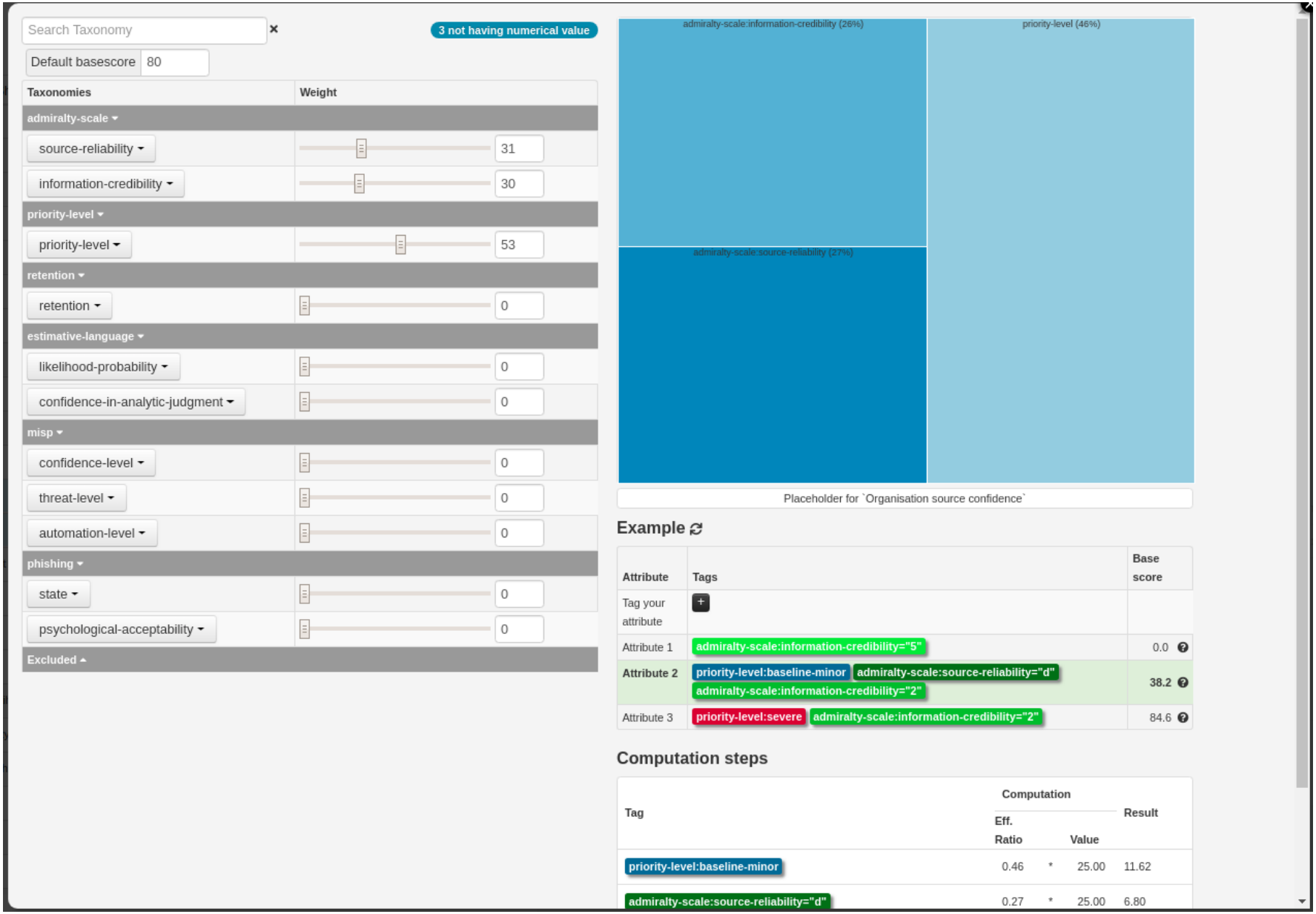
View, update, add, create, delete, enable, export, import

Implementation in MISP: Fine tuning tool



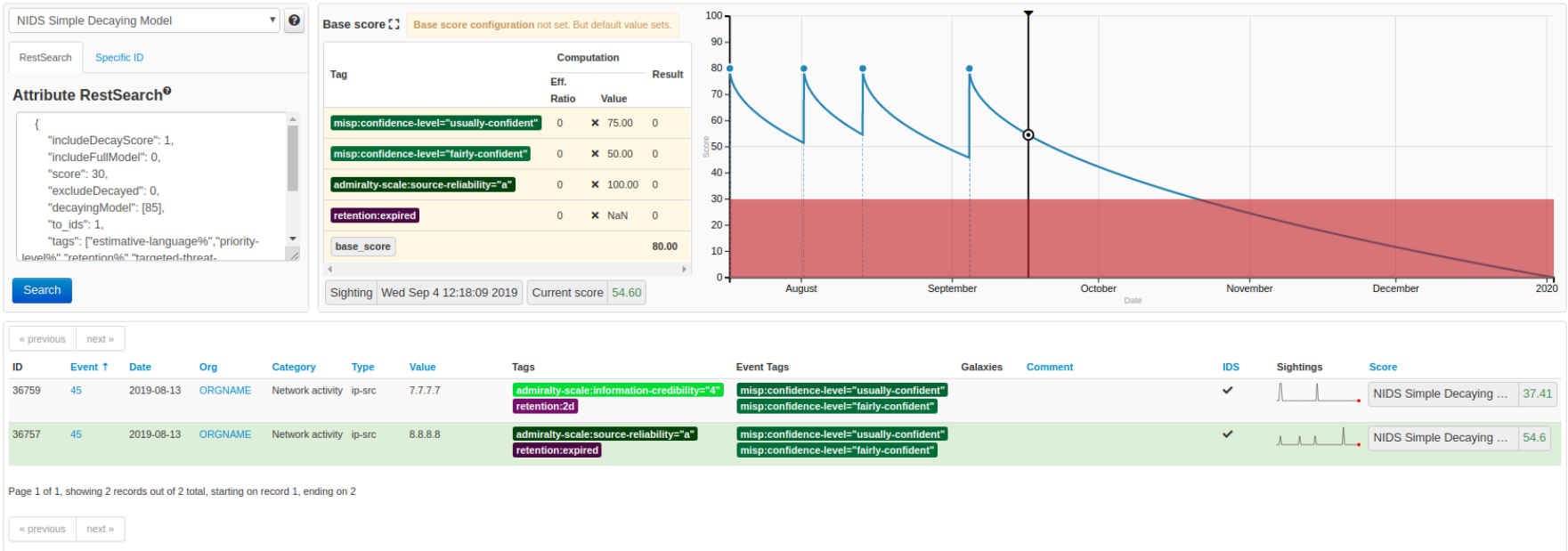
Create, modify, visualise, perform mapping

Implementation in MISP: base_score tool



Adjust Taxonomies relative weights

Implementation in MISP: simulation tool



Simulate *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,
}
```

To sum it all up...

- Massive rise in **user capabilities**
- Growing need for truly **actionable threat intel**
- Lessons learned:
  - **Context is king** - Enables better decision making
  - **Intelligence and situational awareness** are natural by-products of context
  - Don’t lock users into your **workflows**, build tools that enable theirs

Get in touch if you have any questions

- Contact us
  - https://twitter.com/mokaddem_sami
  - https://twitter.com/iglocska
- Contact CIRCL
  - info@circl.lu
  - https://twitter.com/circl_lu
  - https://www.circl.lu/
- Contact MISPProject
  - https://github.com/MISP
  - https://gitter.im/MISP/MISP
  - https://twitter.com/MISPProject