

Fail frequently to avoid disaster or how to organically build a threat intel sharing standard



CIRCL

Computer Incident
Response Center
Luxembourg



MISP
Threat Sharing

Alexandre Dulaunoy

Andras Iklody

TLP:WHITE

Cyber Threat Intelligence
Matters - OASIS / FIRST.org -
Prague 2017

There was never a plan. There was just a series of mistakes.

Robert Caro, journalist.

What the f* is MISP project?

- MISP¹ is a threat information sharing free and open source software.
- MISP has a **host of functionalities** that assist users in creating, collaborating and sharing threat information - e.g. flexible sharing groups, **automatic correlation**, free-text import helper, event distribution and proposals.
- Many export formats which support IDses / IPses (e.g. Suricata, Bro, Snort), SIEMs (eg CEF), Host scanners (e.g. OpenIOC, STIX, CSV, yara), analysis tools (e.g. Maltego), DNS policies (e.g. RPZ)
- MISP project also includes collaborative common vocabularies such as taxonomies, galaxy (e.g. threat-actors or ATT&CK), common objects template and many sub-projects (more than 40 repositories and +200 contributors).

¹<https://github.com/MISP/MISP>

Model of "governance"

- Dictatorship instead of democracy
- Gathering ideas, issues, use-cases, code from the community is key, listen to them but reserve the **right to veto**
 - Prevents malevolent community members from blocking the process/imposing tunnel-visioned ideas
- **Don't wait for the perfect implementation**, start small extend it later
- If the idea doesn't seem suitable for the above, shelf it as soon as possible



Why standardise at all?

- More and more requests from **other tools/vendors to integrate with MISP**
 - Complaints about having to go through a jungle of PHP or Python code to figure out how to do it
- Validation from 3rd parties on the format and overall design
- Describing the scope of the native MISP formats

Development process based on failures

- All ideas need **real-world and practical validation**
- Be willing to throw away features that "sure seemed like a good idea at the time"
- Fail as early as possible (and be proud of your failures)
- Failures can often be used to pinpoint better alternatives
- Format follows the implementation (**code is law**)



Programming, Motherfucker

Do you speak it?

We are a community of motherfucking programmers who have been **humiliated** by software development methodologies for years.

We are tired of *XP, Scrum, Kanban, Waterfall, Software Craftmanship* (aka *XP-Lite*) and anything else getting in the way of...**Programming, Motherfucker.**

We are tired of being told we're socially awkward idiots who need to be manipulated to work in a Forced Pair Programming chain gang without any time to be creative because none of the 10 managers on the project can do...

Programming, Motherfucker.

We must destroy these methodologies that get in the way of...**Programming, Motherfucker.**

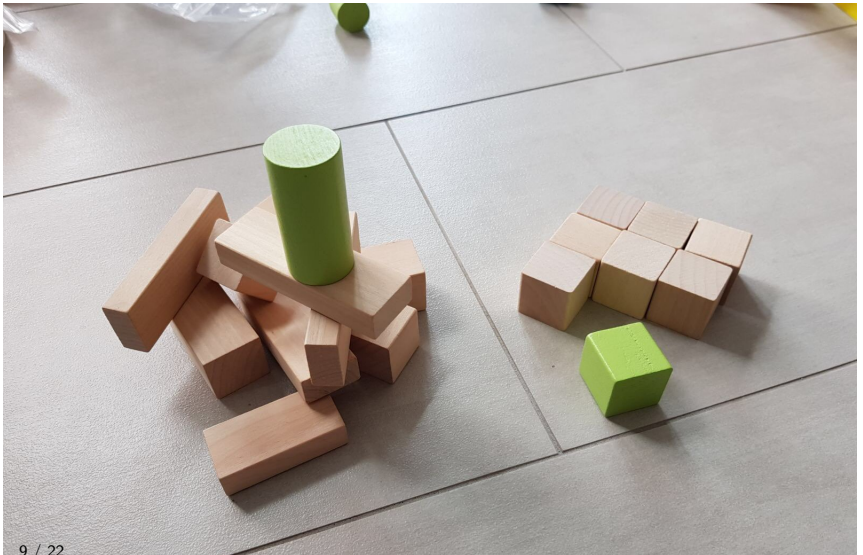
* * * *

Our Values

Staying with theoretical models for too long...

- The same mistakes will be made anyway
- **Piling mistakes on shaky foundations** will be more difficult to undo later
 - Technical reasons (inheritance of the *crap*)
 - Sunk cost fallacy *mistakes seen as failure with any suggestion to rectify it being taboo*
- We generally had two main design goals when it comes to the format:
 - **Design the format, in a way, to be as simple as possible** to be able to map whatever information we want to convey
 - Enhance the format when it's needed instead of planning ahead - **code is law**

Pilling mistakes on shaky foundations (another view)



Scoping the format

- One of the most challenging tasks is having a **clear scope** and a unified vision on what problems we intend to solve
- This can be fluid over time, but the format should stay coherent at any specific point in time with the other components
- Our guiding principle as a **sharing format was to keep complexity levels at a minimum but cover a large spectrum of use-cases**
- We are firm believers that our multi-purpose nature will hinder us at ever being as good at specialised tasks as the relevant specialised formats in the field (Suricata, Bro, Snort, Yara or Sigma)

Designing a standard with sharing in mind (an organic approach)

- The original sharing aspect (in 2011) were quite limited (private flag)
 - If I want to keep it within my organisation, simply set the flag
 - If not set any organisation can see it on the MISP instance
- Utterly simplistic, only worked on communities using a hosted MISP
- **Not known practical cyber security sharing models** known at that time

Designing a standard with sharing in mind (an organic approach)

- The second step of sharing (in late 2012) based on feedback from the previous iteration
- Needed to be extended once communities started self-hosting MISIP **to be able to control the distance of the data-flow**
- Distribution levels
 - Organisation only (private)
 - Community
 - Connected community
 - All

Designing a standard with sharing in mind (going all out)

- Still not covering all use-cases, certain types of users wanting more granularity
- **Extending the current sharing models with a mixed sharing model** via sharing groups (in 2015)
 - Sharing groups (distribution lists)
 - Complex system for persistent and special ad-hoc use-cases (e.g. short-term information exchange)
- Next step: Multiple sharing groups/nested sharing groups

The great failure of free-text tagging

- **Humans can be very creative** especially when they have a playground
- Free-text tagging was a nifty feature in early version of MISP but we underestimated the creativity of the human mind
- "TLP AMBER", "TLPAMBER", "TLP-amber", "TLP:AMBER", "TLP=AMBER" and "TLP/AMBER", "tlp:amber"
- Classifications must be globally used to be efficient. In November 2015, we designed a complete taxonomy system to initially support TLP
- As of Today, we have **more than 40 taxonomies** (from markings, classification taxonomies or even crowdsourced support to allow collaborative analysis)

Taxonomy

- It solved the "creativity issue" but we were only allowing tagging at event level. Attribute level tagging was then introduced in 2016

		Filters: All File Network Financial Proposal Correlation Warnings Include deleted attributes Show c				
<input type="checkbox"/> Date	Org	Category	Type Value	Tags	Comment	
<input type="checkbox"/> 2017-12-03		External analysis	text	Google is constantly working to improve our systems that protect users from Potentially Harmful Applications (PHAs). Usually, PHA authors attempt to install their harmful apps on as many devices as possible. However, a few PHA authors spend substantial effort, time, and money to create and install their harmful app on a small number of devices to achieve a certain goal. This blog post covers Tizi, a backdoor family with some rooting capabilities that was used in a targeted attack against devices in African countries, specifically: Kenya, Nigeria, and Tanzania. We'll talk about how the Google Play Protect and Threat Analysis teams worked together to detect and investigate Tizi-infected apps and remove and block them from Android devices.	<code>osint:source-type="blog-post" x</code> <code>osint:source-type="manual-analysis" x</code> <code>osint:lifetime="perpetual" x</code> <code>osint:certainly="100" x +</code>	
<input type="checkbox"/> 2017-12-03		External analysis	link	https://security.googleblog.com/2017/11/tizi-detecting-and-blocking-socially.html	<code>osint:source-type="blog-post" x</code> <code>osint:source-type="manual-analysis" x</code> <code>osint:lifetime="perpetual" x</code> <code>osint:certainly="100" x +</code>	Google blog post - Tizi: Detecting and blocking socially engineered spyware on Android

Ongoing effort to standardise MISP

- IETF draft document for the MISP core format
- IETF draft documents for the MISP supporting formats
 - Ensuring a separation between the core format and the **extensible and reusable** formats such as taxonomy, galaxy and object.
- Available at <https://github.com/MISP/misp-rfc>

A list of the currently described MISP formats

- MISP core format: basically the exchange format of MISP (Events, Attributes, Objects, Tags, Sharing Groups, Proposals...)
- MISP JSON formats:
 - MISP taxonomies
 - MISP galaxies
 - MISP object-templates

The MISP core format

- Describes the format used to exchange information between MISP instances
- Includes descriptions of all structures that get exchanged between MISPs
 - Events
 - Objects, Object References
 - Attributes, Proposals
 - Tags, Galaxies
 - Organisations

The MISP taxonomy and galaxy formats

- Describes the formats used to create the JSON structures for the respective objects
- Due to the wealth of categorisation/contextual information, used by more and more organisations even outside of MISP (such as Alienvault OTX)
- The standards aim to make life for content creators easier
- Unlike technical information meant for machine ingestion, higher level threat intelligence structure aimed at human analysts can be much more lax in terms of structure
- The format uses a **freely definable key-value store system** to describe data not directly foreseen in the format itself

The MISP object template format

- Since the release of MISP objects, users have started building their own object templates
- These templates are then used to create individual objects based on the pre-defined patterns
- Also includes a vocabulary containing the default relationships to be used for object references and soon galaxy referenced

Theory and practice sometimes clash. And when that happens, theory loses. Every single time.

Linus Torvalds

Q&A

- info@circl.lu (if you want to join the CIRCL MISP sharing community)
- OpenPGP fingerprint: 3B12 DCC2 82FA 2931 2F5B 709A 09E2 CD49 44E6 CBCD
- <https://www.circl.lu/services/misp-malware-information-sharing-platform/>
- <https://github.com/MISP/> - <https://www.misp-project.org/>