Best Practices in Threat Intelligence

MISP Project

Table of Contents

Introduction

The aim of this book is to compile the best practices in threat intelligence analysis.

Whilst this book can be used as a general guide, it is based on the open source threat intelligence platform called MISP to give the reader the most practical and real-world experience.

The best practices described herein are from Information Sharing communities (ISAC or CSIRT) which are regularly using MISP to support their work and sharing practices.

Best Practices

Improving Analysis

> Improvement of the analysis process can range from a simple notification of a false-positive or the correction of a typographic error, all the way up to a complete competitive or counter analysis of the original analysis.

A common difficulty in threat intelligence is to improve existing analyses and especially how to do it efficiently. One of the main questions to ask is:

"What will be the target audience of the improved analysis and the objective thereof?"

The following three answers could come to mind.

1. Informing the original analyst/author (e.g. a security vendor or a CSIRT) about a specific mistake or error which needs to be corrected.
2. Improving an existing analysis by performing a complementary analysis or review which will be shared to and used by another group (e.g. a specific constituent, or a team within your organisation or a member of an ISAC, etc).
3. The end-consumer will be an automaton.

In the **1st** case, MISP includes a mechanism to propose changes to the original creator, a mechanism MISP refers to as proposals. By using proposals, you can propose a change to the value or the context of an attribute (such as a typographic error in an IP address, missing contextual information, type of the information, the category or the removal of an IDS flag). The proposal will be sent back to the original author who can decide to accept or discard it.

The advantages of using the proposal system include the lack of a need to create a new event as well as the process itself being very simple and fast. However, it assumes that the party providing the improvements is willing to lose control over the proposed data. This is pretty efficient for small changes but for more comprehensive changes, especially those that include non-attribute information such as galaxy clusters or objects, the event extension is more appropriate.

Apart from being more suitable for more comprehensive changes, the **2nd** scenario is also a great fit for the extended event functionality, allowing users wanting to provide additional information or an alternate view-point with the opportunity of creating a self-contained event (which can have its own custom distribution rules) that references the original analysis. This information can be shared back to the original author or kept within a limited distribution scope such as a specific sector, a trust group or as internal information for the organisation providing the additional information.

> For more information about the extended event functionality in MISP, the blog post **Introducing The New Extended Events Feature in MISP** includes a lot of details.

In the **3rd** scenario your use-case might be highly automated, e.g. scripted processing of events and attributes via PyMISP and the end-consumer is mainly another automated process, e.g. Intrusion Detection System, 3rd part visualization tool etc. This, for automagic reasons, becomes exponentially unreliable. What is primal in this case is to fully understand what the IDS flag in MISP does and how it impacts attributes. Further on, it is even more important to fully understand the entire tool-chain, cradle-to-grave style. Where does the data come from (cradle) where does it go to (grave) and what processes "touch" the data as it flows through, small diagrams can help tremendously to visualize the actual data-flow. Those diagrams will mostly be of use once unexpected results occur, or other errors appear somewhere in the chain.

What To Share or What Counts As Valuable Information?

Valuable information is a moving concept and depends highly on the goal of the users sharing and/or using the information. A valuable information can also evolve following the capabilities of an organisation.

Contribution comes in various shapes and sizes.

Information which is often distributed within sharing communities are the following:

- Analysis report of a specific threat (such as security vendor report, blog post) which can be Open Source intelligence or come as limited distribution
- Enhanced analysis of an existing report (such as data qualification, competitive or counter analysis)
- A post-mortem analysis of an incident
- Additional information about existing or known threats (such as adversary techniques, new malware samples or complementary discoveries)
- False-positive or false-negative reporting
- Asking for contribution or support from the community (such as "have you seen this threat?" or "do you have more samples?")

By having a look at the object templates or the MISP attribute types, this can help you discover what is actively shared within other communities. If a type or an object template is not matching your data model, you can easily create new ones.

When asking for the support of the community, using a specific taxonomy such as collaborative intelligence to express your needs, will make your request more concise improving your feedback potential and improve automation.

Intelligence Tagging

There are several factors to successful and efficient intelligence sharing. Certainly, one major aspect is the quality of the indicators (or observable depending on the definition you use), stored as attributes within a MISP event itself. However, it does not stop there. Even the most viable information gained by a shared event can render itself complete useless if not classified and tagged accordingly. One feature which enables a uniformed classification is implemented in MISP as tags. Currently, there are two types of tags, which differ in the respective place they are set.

1. You can add tags to an **entire event**. These **tags should be valid for any individual attribute**, thus indicator associated to this specific event.
2. For a more fine-grained specification all of these tags can also be placed at attribute level. This allows the user to put a more detailed and selective view on each attribute.

> ℹ Currently there is no programmatic way that prevents you from not following the 1st rule. Thus human garbage tagging in automation output potentially useless.

> ℹ In future releases there will also be tagging for MISP Objects. Which is, somehow, an intermediate solution for the two prior mentioned options.

> ℹ MISP Objects in its plain concept is a grouping of indicators within one event. These grouped indicators are somehow logically linked together. The specific relationship is described by the individual object type. A simple **file object**, links for example a filename to its observed hash values (md5, sha1, sha256 and many more). This can further be enriched via misp-modules or other plug-ins.

A frequent use-case for placing additional tags on attribute level would be to lower the confidence in certain attributes. If the event is classified with a high confidence tag, some indicators e.g. legit-but-compromised domains or popular filenames should be labeled with a lowered confidence class. There are several real world examples where this or similar attribute specific tagging has proven to be worthwhile.

Most of the tags are organised in dedicated MISP Taxonomies. Those schema dictate how tags should look like and how they are to be applied in certain conditions. There are many general details on this topic which can be read up on in the main MISP Taxonomy GitHub repository. Currently, there are more than 60 different taxonomies available, each of them containing a number of different tags, which are steadily increasing and evolving. There are a lot of advantages in having such a vast variety of tags, e.g. there is one tag for each known associated malware type.

However, this sheer amount of tags can lead to two main concerns, **over-tagging** and **miss-tagging**. Beginners can be overwhelmed with the large number of available tags, and might miss exactly the required taxonomy to properly label the to be shared data. As a site administrator it is thus important to enable the taxonomies that are known to the users on the MISP instance, (or to remote organizations you might sync with).

In MISP making a Taxonomy available is a 2-step process. First you make the taxonomy available and then you can either decide to enable all the individual tags in the taxonomy or cherry-pick only the relevant ones for your use-case. (The Vocabulary for Event Recording and Incident Sharing (VERIS) has well over 1990 tags, and perhaps you are only interested in the sub-set `veris:action:error:variety`).

Over-tagging in most cases only leads to an overwhelming visual appearance. Miss-tagging, however, is a critical step into mis-usage of shared data. The best and most devastating example would be the miss classification of an event. In dedicated and private sharing groups it is quite usual to share intelligence labeled as „for your company only“. This data must not leave the boundaries of this virtual border of the recipient’s firm.

To prevent this kind of mistake, the traffic light protocol (aka TLP) and its respective taxonomy can be used and thus complementing the mitigation in the note below.

One mitigation the scenario of mis-classified data, would be to use the MISP warninglists (or MISP noticelists) as a canary. Whilst not ideal and far from a defacto solution to catch all issues, it would be a good-enough-yet-coarse way of detection.

There are multiple solutions to solve the issue of missing additional information about the shared content. One of them is the following list of tags which are deemed to be the minimal subset at the start of any event or the individual attributes. sharing platform. The list below is in order of importance.

1. **TLP-Tags**: TLP utilizes a simple four color schema for indicating how intelligence can be shared.
2. **Confidence-Tags**/**Vetting State**: There are huge differences in the quality of data, whether it was vetted upon sharing. As this means that the author was confident that the shared data is or at least was a good indicator of compromise.
3. **Origin-Tags**: Describes where the information came from, whether it was in an automated fashion or in a manual investigation. This should give an impression how value this intelligence is, as manual investigation should supersede any automatic generation of data.
4. **PAP-Tags**: An even more advanced approach of data classification is using the Permissible Actions Protocol. It indicates how the received data can be used to search for compromises within the individual company or constituency.

The full list of available taxonomies can be found **here**.

Expressing confidence/estimative probability in an analysis

Expressing the confidence or the lack of it in an analysis is a critical step to help a partner or a third-party to check your hypotheses and conclusions.

Analysis or reports are often shared together with technical details, but often lack the associated overall confidence level. To ascertain this confidence level you can use for example the MISP MISP Taxonomies called admiralty-scale and/or estimative-language. This is a very human way to describe either globally an event or individual indicators of an event, with a set of easy to read human tags. (e.g: admiralty-scale:source-reliability="a/b/c...", estimative-language:likelihood-probability="almost-no-chance", estimative-language:confidence-in-analytic-judgment="moderate") Generally it is good practice to do this globally for the event as this will enrich the trust/value if set. Using this in an automated way is also possible but without human intervention, or AI that actually works, not recommended. Also, on events with hundreds of attributes this is cumbersome and perhaps unfeasible and will just frustrate operators. The obvious side-effect of this approach is that automation will be the overall benefactor too upping the trust on that level too.

Thus, adding confidence or estimative probability has multiple advantages such as:

- Allow receiving organisations to filter, classify and score the information in an automated way based on related tags
- Information with low-confidence can still be shared and reach communities or organisations interested in such information without impacting organisations filtering out by increased confidence level
- Support counter analyses and competitive analyses to validate hypotheses expressed in original reporting
- Depending on source organisation, have an affirmative that some HumInt has one into the sharing process

Complement analysis with contrary evidences is also very welcome to ensure the original analysis and the hypotheses are properly evaluated.

MISP Taxonomies contain an exhaustive list of confidence levels including words of estimative probability or confidence in analytic judgment.

threat-intelligence.eu includes an overview of the methodologies and process to support threat intelligence.

How to track and keep the state of an analysis

Having a workflow to follow, and be able to refer to, is something useful for the analyst as well as for other people reading or relying on the analysis.

Keeping track of the advancement of an analysis, of what has been done or still needs to be done, is important in order not to forget anything on either side and to ensure work is not performed redundantly by accident. It is essential to have a method to keep these information clear and concise.

One of the possible methodologies is to use tags to mark the information and convey the current state of an analysis.

For instance the MISP Workflow [Taxonomy] allows the user to describe the state of an analysis, as `complete` or `incomplete`. Moreover, it can be used to clearly specify what still needs to be done using the `todo` tags. The workflow taxonomy is separated into two parts. One part is related to the actions to be done (`todo`) and the other part is about the current state of the analysis(`state`) such as `incomplete`, `draft` or `complete`.

For more information on the MISP Workflow Taxonomy, feel free to read the Workflow taxonomy cheat sheet.

How to classify information

> Classifying information is something that has proven being very useful in lots of domains, including Threat Intelligence, as it helps assessing the main information very quickly. Moreover, it can help to build correlations between events or reports, allowing analysts to better understand threat actors.

The first tool we can use to classify information are tags and taxonomies

1. Tags can be used to describe how the information can be shared, using the tlp (Traffic Light Protocol) taxonomy, in order to prevent information leaks.
2. They can also be used to describe the source where information came from.
3. Many taxonomies allow the user to further explain the kind of threat.

- [MISPGalaxies] (ATT&CK matrix)
- Comments

Authors and Contributors

- Alexandre Dulaunoy
- Andras Iklody
- Steve Clement

Glossary

MISP Glossary

This glossary is meant as a quick lookup document in case of any need of clarification of any threat sharing, threat-intel lingo. Be careful when adding terms to the glossary. Adding a generic term like: **MISP** will prevent terms like **MISP noticelist** to be addded. As a matter of definition please use the singular for any terms. In case you use any CCBYSA licensed content, or other pieces that are subject to licensing, make sure to add it as a by-line at the end of the mention.

ISAC

Information Sharing and Analysis Center

MISP

MISP - Open Source Threat Intelligence Platform & Open Standards For Threat Information Sharing

MISP Modules

MISP modules are autonomous modules that can be used for expansion and other services in MISP. MISP modules GitHub Repository

MISP warninglists

MISP warninglists are lists of well-known indicators that can be associated to potential false positives, errors or mistakes. MISP warninglists GitHub Repository

MISP noticelists

Notice lists to inform MISP users of the legal, privacy, policy or even technical implications of using specific attributes, categories or objects. MISP noticelist GitHub Repository

MISP Taxonomies

Taxonomy is the practice and science of classification. The word is also used as a count noun: a taxonomy, or taxonomic scheme, is a particular classification. The word finds its roots in the Greek language τάξις, taxis (meaning 'order', 'arrangement') and νόμος, nomos ('law' or 'science'). For more details on taxonomies and classification the documentation. Partial source "Taxonomy_(general)" - CCBYSA. There is a Python module available to work with Taxonomies in a Pythonic way called PyTaxonomies. MISP taxonomies GitHub Repository

MISP Sightings

Basically, sighting is a system allowing people to react on attributes on an event. It was originally designed to provide an easy method for user to tell when they see a given attribute, giving it more credibility.

MISP Objects

MISP objects are used in MISP (starting from version 2.4.80) system and can be used by other information sharing tool. MISP objects are in addition to MISP attributes to allow advanced combinations of attributes. The creation of these objects and their associated attributes are based on real cyber security use-cases and existing practices in information sharing. The objects are just shared like any other attributes in MISP even if the other MISP instances don’t have the template of the object. The following document is generated from the machine-readable JSON describing the MISP objects. MISP objects GitHub Repository More

API

MISP makes extensive use of its RESTful API (Application programming interface) both internally and provides an external API for automation, synchronisation or any other tasks requiring a machine to machine interface. In general terms, it is a set of clearly defined methods of communication between various software components. A good API makes it easier to develop a computer program by providing all the building blocks, which are then put together by the programmer. An API may be for a web-based system, operating system, database system, computer hardware or software library. The de-facto standard for talking to MISP via an API is PyMISP. Partial source "API" - CCBYSA.

RESTful

Representational state transfer (REST) or RESTful web services are a way of providing interoperability between computer systems on the Internet. REST-compliant Web services allow requesting systems to access and manipulate textual representations of Web resources using a uniform and predefined set of stateless operations. Other forms of Web services exist which expose their own arbitrary sets of operations such as WSDL and SOAP. Source "REST" - CCBYSA.

PyMISP

PyMISP is a Python library to access MISP platforms via their REST API. PyMISP allows you to fetch events, add or update events/attributes, add or update samples or search for attributes.

IDS

An IDS flag on an attribute allows to determine if an attribute can be automated (such as being exported as an IDS ruleset or used for detection). If the IDS flag is not present, the attribute can be useful for contextualisation only.

IOC

Indicator of compromise (IOC or IoC) is an artefact observed on a network or in an operating system or information channel that could reference an intrusion or a reference to a technique used by an attacker. IoCs are a subset of indicators.

Attribute

Attributes in MISP can be network indicators (e.g. IP address), system indicators (e.g. a string in memory) or even bank account details.

Observable

Obserbables are essentially the same as (MISP) attributes.

Site Admin

As an admin (not to be confused with Org Admin), you can set up new accounts for users, edit user profiles, delete them, or just have a look at all the viewers' profiles. Site admins have access to every administrator feature for all the data located on the system including global features such as the creation and modification of user roles and instance links. You will also see all other organisations connected or setup in the instance. The site admin can be considered as a super-user of a MISP instance.

Org Admin

Organisation admins (Org Admin) are restricted to executing site-admin actions exclusively within their own organisation’s users only. They can administer users, events and logs of their own respective organisations.

OSINT

Open-source intelligence (OSINT) is data collected from publicly available sources to be used in an intelligence context.[1] In the intelligence community, the term "open" refers to overt, publicly available sources (as opposed to covert or clandestine sources). It is not related to open-source software or public intelligence. OSINT under one name or another has been around for hundreds of years. With the advent of instant communications and rapid information transfer, a great deal of actionable and predictive intelligence can now be obtained from public, unclassified sources. Source "Open-source intelligence" - CCBYSA.