# Contributing to MISP Project

MISP project is a large free software project composed of multiple sub-projects which are contributed by different contributors who are generally active users of the MISP project. MISP project fully supports the [Contributor Covenant Code of Conduct](https://github.com/MISP/MISP/blob/2.4/code\_of\_conduct.md) to foster an open and dynamic environment for contributing and the exchange in the threat intelligence and information exchange field.

The MISP roadmap is mostly based on the user communities (e.g. private communities, CSIRTs communities, security researchers, ISACs - Information Sharing and Analysis Center, security providers, governmental or military organisations) relying on MISP to perform their duties of information sharing and analysis. If you see an existing issue which covers your needs, don't hesitate to comment on it.

Participating in the MISP project is easy and everyone can contribute following their own ability:

## Reporting a bug or an issue, suggesting features

Reporting a bug or an issue, suggesting a new feature in the [MISP core](https://github.com/MISP/MISP/issues), [misp-modules](https://github.com/MISP/misp-modules/issues), [misp-book](https://github.com/MISP/misp-book/issues), [misp-taxonomies](https://github.com/MISP/misp-taxonomies/issues), [misp-galaxy](https://github.com/MISP/misp-galaxy/issues) or any of the projects within the [MISP project org](https://github.com/MISP/). Don't hesitate to add as much information as you can, including the version of MISP which you are running, screen-shots with annotation, suggested features and steps on how to reproduce an issue. Don't hesitate to search the existing open issues and comment on existing ones. This is an indicator for us regarding the priority of certain features

and how important these are to the users.

## Reporting security vulnerabilities

Reporting security vulnerabilities is of great importance for us, as MISP is used in multiple critical infrastructures. In the case of a security vulnerability report, we ask the reporter to directly report to [CIRCL](https://www.circl.lu/contact/), encrypting the report with the GnuPG key: CA57 2205 C002 4E06 BA70 BE89 EAAD CFFC 22BD 4CD5. We usually fix reported and confirmed security vulnerabilities in less than 48 hours, followed by a software release containing the fixes within the following days. If you report security vulnerabilities, don't forget to tell us if and how you want to be acknowledged and if you already requested CVE(s). If not, we will request the CVE directly.

As one of the critical user-bases of MISP consists of the CSIRT community, it is our duty to clearly state which bug could be potentially abused and could have a security impact on a deployed MISP instance. CVE assignment is performed even for minor bugs having some possible security impact. This allows users using MISP instances in their environment to understand which bugs could have an impact on their security. We firmly believe that, even though unfortunately it is often not regarded as common practice in our industry, being as transparent as possible about vulnerabilities, no matter how minor, is of absolute crucial importance. At MISP-project, we care about the security of our users and prefer to have a high number of published CVEs than to a few swept under the rug.

## MISP core

If you want to contribute to the [MISP core](https://github.com/MISP/MISP) project:

- First fork the [MISP core project](https://github.com/MISP/MISP)

- Branch off from 2.4 (2.4 branch is the main branch of development in MISP) `git checkout 2.4`

- Then create a branch for your own contribution (bug fixes, enhancement, new features) by typing `git checkout -b fix-glossy-user-interface`

- Work on your fix or feature (only work on that, avoid committing any debug functionalities, testing or unused code)

- Commit your fix or feature (and sign it with GnuPG - if you have a GnuPG key) with a meaningful commit message as recommended in [A Note About Git Commit Messages](http://tbaggery.com/2008/04/19/a-note-about-git-commit-messages.html).

- MISP uses [gitchangelog](https://github.com/vaab/gitchangelog/blob/master/src/gitchangelog/gitchangelog.rc.reference) to generate changelog, so it's recommended to use `new:` for new features, `fix:` when it's a bug-fix or `chg` when it's re-factoring or clean-up..

- Push and then open a pull-request via the GitHub interface.

If you never did a pull-request, there is [The beginner's guide to contributing to a GitHub project](https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/)

Some recommendations to get your PR merged quickly:

- Small gradual changes are preferred over big and complex changes.

- If the changes require some documentation changes, a pull-request on [misp-book](https://github.com/MISP/misp-book) is strongly recommended.

- If the commit message contain all the information regarding the changes, it's easier for the maintainer to do the review.

- Avoid committing sensitive information, debugging code or unrelated code in the PR.

## MISP taxonomies

If you cannot find an existing taxonomy fitting your needs, you can extend an existing one (especially the ones originated from the MISP project) or create a new one.

Create a JSON file describing your taxonomy as triple tags (e.g. check an existing one such as the [Admiralty Scale](https://github.com/MISP/misp-taxonomies/tree/master/admiralty-scale)) taxonomy, create a directory matching your name space, put your machinetag file in the directory and create a pull request. That's it. Everyone can benefit from your taxonomy and it can be automatically enabled in information sharing tools such as [MISP](https://www.github.com/MISP/MISP).

We strongly recommend to validate the JSON file using [jq](https://github.com/MISP/misp-taxonomies/blob/master/jq\_all\_the\_things.sh) before doing a pull-request. We also strongly recommend to run [the validator](https://github.com/MISP/misp-taxonomies/blob/master/validate\_all.sh) to check if the JSON validates the schema.

For more information, see the presentation slides on "[Information Sharing and Taxonomies Practical Classification of Threat Indicators using MISP](https://www.circl.lu/assets/files/misp-training/3.2-MISP-Taxonomy-Tagging.pdf)" given at the last MISP training in Luxembourg.

## MISP galaxy

In the world of threat intelligence, there are many different models or approaches to order, classify or describe threat actors, threats or activity groups. We welcome new ways of describing threat intelligence as the galaxy model allows to reuse the ones you use or trust for your organization or community.

Fork the project, update or create a galaxy or clusters and make a pull-request.

We strongly recommend to validate the JSON file using [jq](https://github.com/MISP/misp-galaxy/blob/master/jq\_all\_the\_things.sh) before doing a pull-request. We also strongly recommend to run [the validator](https://github.com/MISP/misp-galaxy/blob/master/validate\_all.sh) to check if the JSON validates the schema.

## Building software compatible with MISP formats and improving MISP formats

[MISP formats](https://github.com/MISP/misp-rfc) are open and free standards. We invite software developers to use the native MISP format for exchanging threat intelligence and support information sharing/analysis. The MISP formats are simple JSON formats implemented in various software including the MISP core application along with various libraries such as [PyMISP](https://github.com/MISP/PyMISP).

If you want to contribute to the MISP formats (which are actively based on the MISP core implementation), feel free to open an issue at [misp-rfc](https://github.com/MISP/misp-rfc).