## Reporting Security Vulnerabilities

If you think that you have found a security vulnerability, please email security@mautic.com with as much detail as possible. The core team will review the vulnerability and if found applicable, will create the patch in a private repository. The vulnerability will be disclosed once the patch has been included into a release.

## Contributing Code

Development is open and available to any member of the Mautic community. All fixes and improvements are done through pull requests to the code. This code is open source and publicly available.

### Developer Documentation

Developer documentation is available at [https://developer.mautic.org](https://developer.mautic.org). To add additions or corrects to the documentation, submit Issues or Pull Requests against [https://github.com/mautic/developer-documentation](https://github.com/mautic/developer-documentation).

### Core Feature Development Procedures

Pull Requests with additional features should be created with the Mautic Core goals in consideration. Any features that are created for core that don’t follow the overall goals may not be included.

In addition to following the general direction of the development goals, the pull request code must be well-formed following coding standards and guidelines. If you wish to target a specific release version number for the feature, its best to make the pull request early so any feedback from the core team can be implemented and adequate testing can be performed.

Features that are determined not to fit within the direction of the Mautic Core goals are more than welcome to be created as plugins instead.

### Code Contribution Requirements

#### Code Standards

Mautic follows [Symfony's coding standards](http://symfony.com/doc/current/contributing/code/standards.html) by implementing pre-commit git hook running [php-cs-fixer](https://github.com/friendsofphp/php-cs-fixer), which is installed and updated with `composer install`/`composer update`.

All code styling is handled automatically by the aforementioned git hook. In case if you setup git hook correctly (which is true if you ever run `composer install`/`composer update` before creating a pull request), you can format your code as you like - it will be converted to Mautic code style automatically.

#### Automated Tests

All code contributions should include adequate and appropriate unit tests using [PHPUnit](https://phpunit.de/manual/5.7/en/index.html) and/or Symfony functional tests ([https://symfony.com/doc/2.8/testing.html](https://symfony.com/doc/2.8/testing.html)). Pull Requests without these tests will not be merged.

#### Pull Request Description

When creating a new Pull Request, the description template should be filled appropriately in detail. Any Pull Request that does not have an appropriate description will not be considered for merge.

#### Documentation

Each new feature should include a reference to a pull request in our [End User Documentation](https://github.com/mautic/documentation) repository or [Developer Documentation](https://github.com/mautic/developer-documentation) repository if applicable.

## Core Development Rules

Pull requests and code submissions are decided upon by the release leader and the core team. When a decision is not clearly evident then the following voting process will be implemented.

### Voting Policy

Votes are cast by all members of the core team. Votes can be changed at any time during the discussion. Positive votes require no explanation. A negative vote must be justified by technical or objective logic. A core team member cannot vote on any code they submit.

### Merging Policy

The voting process on any particular pull request must allow for enough time for review by the community and the core team. This involves a minimum of 2 days for minor modifications and minimum of 5 days for significant code changes. Minor changes involve typographical errors, documentation, code standards, minor CSS, javascript, and HTML modifications. Minor modifications do not require a voting process. All other submissions require a vote after the minimum code review period and must be approved by two or more core members (with no core members voting against).

### Core Membership Application

Core team members are based on a form of meritocracy. We actively seek to empower our active community members and those demonstrating increased involvement will be given everything needed for their continued success.

### Core Membership Revocation

A Mautic Core membership can be revoked for any of the following reasons:

- Refusal to follow the rules and policies listed herein

- Lack of activity for the previous 6 months

- Willful negligence or intent to harm the Mautic project

- Upon decision of the project leader

Revoked members may re-apply for core membership following a 12 month period.