# How to contribute

Contributions are always welcome, however, to keep things consistent, please review the following guidelines.

## Setting up your environment

Ensure that you have Ruby >= 2.4.4 installed on your system and then install all required dependencies by opening a command prompt / terminal in the WPXF folder and running ```bundle install```.

If bundler is not present on your system, you can install it by running ```gem install bundler```.

## Update unit tests

If you change a core piece of functionality (i.e. in ```lib/\*```) then ensure the corresponding unit tests in the ```spec``` folder are updated.

For more information on writing unit tests with RSpec, see https://relishapp.com/rspec

## Ensure RuboCop approves

Unless there's good reason, there should be no [RuboCop](https://github.com/bbatsov/rubocop) warnings for any code you submit a pull request for. There have been changes to the [RuboCop](https://github.com/bbatsov/rubocop) configuration for this project to disable some rules and to increase the boundaries of some (such as line lengths), so when testing your files, ensure you're doing so using the ```.rubocop.yml``` found in the root directory.

## Provide instructions for testing new exploits and auxiliary modules

If you're sending a pull request for a new auxiliary or exploit module, provide some basic information on how to test it. This should include:

- If the module targets a plugin or theme, provide a link to the vulnerable plugin / theme

- If the module targets a specific version of WordPress, be clear which version or range it targets

- Provide links to reference material if available (e.g. WPVulnDB link)

## Target the development branch

When opening a pull request, compare with the ```development``` branch, rather than ```master```. The master branch is aimed at being equal to the latest stable release; meaning all staged changes need to go into the ```development``` branch.