

Security Issues in WordPress

As long as people write code, there will be bugs and errors in every application. Some bugs are critical, while others are trivial. In order to identify bugs in software, open-source projects like WordPress require users' assistance.



In WordPress project, all types of feedback are reported the same way, whether they are featured requests, or genuine bugs. Read Contributing to WordPress to learn the basics of WordPress.

How to Report Security Issues

If you think you have found a bug in WordPress release, please read the Security FAQ to know how to report it.

To minimize the public damage and prepare a bugfix, it is a common practice to let WordPress know developers about a security problem first before posting a

Saturday, April 20, 2024

The author of the following page

<https://dev.bestwebsoft.com/blog/security-issues-in-wordpress/> is
Catherine.



report.

Report about Bugs in Themes and Plugins

Instructions provided in this article do not apply to bugs in themes and plugins! They apply to bugs in WordPress core only. So do **NOT** report a bug using the procedures in this article if you find a bug in a Plugin or Theme you are using! Plugins have a separate bug tracking system from WordPress core, since they are stored in the WordPress Plugin Repository. To use that system, please see other instructions.

For themes and plugins, which are not located in the official repository, see the documentation in the theme/plugin instructions. If there are no instructions available, you will need to contact theme/plugin` authors directly.

How to Prepare a Bug Report

The process of reporting and resolving a WordPress bug has several steps:

1. You find a bug in WordPress core.
2. Check out *Before You Report a Bug*, to make sure it is actually a bug.
3. If it really is a bug, see *Reporting a Bug*, and submit a *ticket* or a bug report.
4. See *Bug Resolutions*. If a WordPress developer confirms that the bug does actually exist, they will mark it in the ticket.
5. WordPress developers will start fixing the bug. The developer may choose to click on the *Accept ticket* option to take direct responsibility for the bug.
6. Volunteers and Members of WordPress development group will test the fixed file to check if it is correct, and does not break anything else.
7. Then, the developers will *commit* the fixed files to the SVN repository.

Saturday, April 20, 2024

The author of the following page

<https://dev.bestwebsoft.com/blog/security-issues-in-wordpress/> is Catherine.



8. The bug ticket status will be set to *closed* and the resolution to *fixed*.

Bug Reporting Details

Before Reporting a Bug

There is a good chance your bug has already been reported, since there is a lot of users in WordPress. Hence, it is very important to check this before submitting the ticket. Also, if you are a beginner in WordPress, it will be useful to discuss the issue with more experienced users. Please, see the information below.

1. Here is a debugging file to make sure the bug is actually coming from WP Core code. Paste it to your *wp-content/mu-plugins* folder (create it if it doesn't exist).
2. Using Query or Search, search for your bug or feature request in Trac. Please, do not report a duplicate bug if your issue has already been reported. If your bug is similar, but not the same as another issue, you can add a note to that similar issue.
3. You can post a question on WordPress Support Forum or discuss your issue on *#wordpress* IRC channel, if you want to discuss a bug before reporting it in Trac (for instance, to ensure it is really a core bug, not caused by a theme or plugin).

How to Report a Bug

The official WordPress bug tracker is called Trac. It uses Trac - the open source bug tracking software. To create a good bug report in Trac, please follow these steps:

1. See Before You Report a Bug, and verify that your bug is appropriate to be reported.
2. Read Trac Ticket documentation and How to Report Bugs Effectively.
3. Using your support forum username and password, log in to WordPress

Saturday, April 20, 2024

The author of the following page

<https://dev.bestwebsoft.com/blog/security-issues-in-wordpress/> is

Catherine.



Trac. The developers may need more information about your bug.

4. To find the bug reporting page, click New Ticket in Trac.
5. Fill in the [#Ticket_Fields].
6. See the preview and click **Submit Ticket**.

How to find a Bug to Fix

Here are some suggestions on finding bugs to fix:

- Look through Trac reports for bugs that look interesting, Needs Patch Report on Trac and Lacks Attachment Report on Trac reports.
- Ask how you can help here wp-hackers mailing list.
- Read WordPress Bug Hunts and subscribe either to wp-testers, or wp-hackers mailing list. Also, join the wp-trac email list to see the discussions about all Tickets.

How to Patch Bugs

If you would like to help in the development of WordPress, and you are familiar with MySQL and PHP, please follow these steps to patch WordPress bugs:

1. Find a bug to fix in Trac. To learn how to do it, read Finding Bugs to Fix first
2. Using the username and password you have, connect to the WordPress Subversion (SVN) Repository and find out how to fix the bug there.
3. Verify that the bug has been fixed, and test it to be sure you haven't broken anything else.
4. Create a *patch file* with your fix. For detailed instructions, see How to Patch WordPress by Owen Winkler, Mark Jaquith Toolbox (for Linux/Mac) and Westi Blog (Windows).
5. Using the Trac *Attach file* button, upload your file to the ticket and add *has-patch* to the keywords.

Saturday, April 20, 2024

The author of the following page

<https://dev.bestwebsoft.com/blog/security-issues-in-wordpress/> is
Catherine.



Patch requirements:

- it should be written according to the coding standards.
- it should be made against the root WordPress directory and against the latest code in the SVN repository.

Ticket Fields

- **Brief Summary.** Prepare a brief but informative summary, since it will be displayed in search results.
- **Description.** Prepare a full description of your bug, including a description of the problem, and why this problem is worth being corrected, as well as an example of the bug in action (i.e. a URL). In addition, include information about your WordPress version, web server software, operating system, MySQL version, PHP version. The better description you have, the better is the effectiveness of your ticket.
- **Ticket Properties**
- **Priority.** This option may be read-only, since the core developers usually rank the bug's priority. It means how urgent your bug is.
- **Component.** Choose the WordPress component, where the bug was found.
- **Severity.** Decide on how critical you consider the issue to be, and select a severity.
- **"Assign to".** This is optional if you are familiar with the developer, who is responsible for the code where you find the bug.
- **Version.** WordPress version where the bug was found.
- **Keywords.** Read Trac Keywords, there are some standard keywords to display your bug` status.
- **CC.** Enter your Trac username here if you want to stay tuned, and get

Saturday, April 20, 2024

The author of the following page

<https://dev.bestwebsoft.com/blog/security-issues-in-wordpress/> is

Catherine.



information and updates on the bug.

Types of Tickets

- A bug. If there are no expected actions.
- Enhancement. When the current action is improved or modified.
- Feature. If a completely new action takes place.

Keywords in Trac

There is a number of commonly used keywords in Trac, which should not be used as generic 'tags:

- **reporter-feedback**. A response from the reporter is needed.
- **dev-feedback**. A response from trusted members of the development community or a core developer.
- **ui-feedback** and **ux-feedback**. If you need a response from the core team.
- **rtl-feedback**. Feedback with regards to RTL support is needed
- **2nd-opinion feedback**. An opinion about the problem from an outside person
- **has-patch**. When you attach a proposed solution for review.
- **needs-patch**. If the proposed solution doesn't work.
- **needs-ui**. When the ticket requires changing several elements.
- **needs-refresh**. The patch needs to be re-submitted and merged.
- **needs-testing**. A couple of persons are needed to test the fix.
- **needs-unit-tests**. The ticket has been tested, found to be desirable to solve, and some unit tests are needed.
- **needs-docs**. Some documentation for the code is needed.
- **needs-codex**. Codex documentation needs to be expanded or updated.
- **commit**. The patch is ready to be added to WordPress core files.

Saturday, April 20, 2024

The author of the following page

<https://dev.bestwebsoft.com/blog/security-issues-in-wordpress/> is

Catherine.



- **early.** The ticket has a high priority and should be solved early in the next release.
- **i18n-change.** It is used to track the potential string changes.
- **close.** The ticket will be closed.

Bug Resolutions

When a ticket was closed, it has one of status designations below:

- **duplicate.** If your bug is a duplicate of the another ticket.
- **invalid.** If it has been decided that it is the intended action, not a bug.
- **worksforme.**
- **wontfix.** If the developers decided that your bug is a feature request that should not be in the core.
- **maybelater.**
- **fixed.** If the bug is fixed.

Your participation may be required during the process. Please, be ready to help the developers in resolving the issue.

“Levels” of Developers

- lead developer, read Development Team.
- core committer: a developer with permanent commit access
- guest committer: a developer with temporary commit access
- core contributor: a person without commit access but who is actively involved in Core development.

Saturday, April 20, 2024

The author of the following page

<https://dev.bestwebsoft.com/blog/security-issues-in-wordpress/> is
Catherine.