

PHP Execution in Uploads

What This Means

This finding indicates that PHP files can be executed within upload or storage directories.

These directories are typically intended for file storage, not code execution.

Why It Matters

Upload directories (such as `/wp-content/uploads/`) are commonly writable by the application.

If PHP execution is allowed in these locations, attackers may:

- upload malicious scripts
- execute code through vulnerable upload points
- gain unauthorized access or control

This is a common method used in real-world WordPress attacks.

How Steel Security Detects This

Steel Security checks whether PHP files placed in upload directories can be executed.

If execution is possible, the directory is flagged as a risk.

How to Fix It

To resolve this issue:

- disable PHP execution in upload and storage directories
- apply server-level rules to prevent script execution
- use Steel Security hardening controls related to execution restrictions

What to Expect After Fixing

After applying protections:

- PHP files in upload directories will not execute
 - uploaded scripts will be treated as files, not code
 - your site will be significantly more resistant to upload-based attacks
-

How to Verify

To verify the fix:

1. upload or place a test PHP file in an upload directory
2. attempt to access it via browser
3. confirm that execution is blocked

Expected results include:

- file download instead of execution
 - or a blocked request (e.g., 403 Forbidden)
-

Common Causes

- default server configuration allowing execution
 - missing directory-level restrictions
 - insecure file upload handling
-

Best Practices

- never allow PHP execution in upload directories
 - treat storage locations as non-executable
 - regularly review upload behavior and permissions
 - combine with other execution and file protection controls
-

Related

- [Prevent Execution in Sensitive Directories](#)
- [Block Direct PHP Access](#)
- [Restrict Sensitive Endpoints](#)

Revision #1

Created 2026-04-04 18:46:58 UTC by Jason Wassing

Updated 2026-04-04 18:46:58 UTC by Jason Wassing