Prevent Execution in Sensitive Directories

What This Does

This protection prevents PHP execution within directories that should only store data, not executable code.

It helps ensure that uploaded or stored files cannot be used to run malicious scripts.

Why It Matters

Certain directories are intended for storage, such as:

- `/wp-content/uploads/`
- cache directories
- temporary or backup locations

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

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

Blocking execution in these directories significantly reduces this risk.

When to Apply It

This protection is strongly recommended for all production websites.

Apply it when:

- your site allows file uploads
- you use media libraries or file storage

- you want to prevent execution of uploaded content

When to Be Cautious

Be cautious when:

- legitimate PHP execution is required in these directories (rare)
- custom applications rely on dynamic execution in storage locations

Always test after applying.

How Steel Security Applies This

Steel Security applies server-level rules to prevent PHP execution in sensitive directories.

This may include:

- disabling PHP processing in upload directories
- restricting execution in storage or cache paths
- applying directory-specific rules

The exact implementation depends on your server environment.

What to Expect After Applying

After applying this protection:

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

How to Verify

To verify the protection:

1. Upload or place a test PHP file in a protected directory

2. attempt to access it via browser
3. confirm that execution is blocked

Expected results include:

- file download instead of execution
- or access denied (e.g., 403 Forbidden)

How to Revert (Rollback)

To revert this protection:

1. Navigate to the hardening section in Steel Security
2. disable or adjust the control
3. confirm the change
4. re-test directory behavior

Common Issues

Uploaded Files Do Not Behave as Expected

- verify file types and handling
- confirm no legitimate PHP execution is required

No Visible Change

This is expected if:

- no executable files are present
- the environment was already secure

Rule Not Applied

- verify server supports directory-level restrictions
- check for conflicting configuration
- confirm rules are active

Best Practices

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

Related

- Block Direct PHP Access
- Protect Configuration Files
- Protect Backup Files
- Safe Rollback Practices

Revision #1
Created 2026-04-04 18:49:18 UTC by Jason Wassing
Updated 2026-04-04 18:49:19 UTC by Jason Wassing