Block Direct PHP Access

What This Does

This protection prevents direct access to PHP files that are not intended to be accessed via a browser.

It ensures that internal scripts cannot be executed outside of normal WordPress workflows.

Why It Matters

Many PHP files within a WordPress site are meant to be included internally, not accessed directly.

If these files are accessible, attackers may:

- execute scripts in unintended ways
- bypass application logic
- probe for vulnerabilities

Blocking direct access reduces this risk.

When to Apply It

This protection is recommended for most production websites.

Apply it when:

- your site contains custom or plugin PHP files
- you want to prevent unauthorized script execution
- you are reducing attack surface

When to Be Cautious

Be cautious when:

- certain PHP files are intentionally accessed directly
- integrations rely on direct script access
- custom endpoints are implemented outside WordPress routing

Test thoroughly after applying.

How Steel Security Applies This

Steel Security restricts direct access to PHP files using server-level rules.

This may include:

- blocking direct requests to specific PHP files
- allowing execution only through WordPress entry points
- applying rules to common sensitive locations

The exact implementation depends on your server environment.

What to Expect After Applying

After applying this protection:

- direct access to restricted PHP files will be blocked
- unauthorized execution attempts will fail
- normal site functionality should remain unchanged

How to Verify

To verify the protection:

1. Attempt to access a PHP file directly via URL
2. Confirm that access is denied (e.g., 403 Forbidden)

Ensure:

- legitimate site functionality still works
- restricted files are not accessible directly

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 affected functionality

Common Issues

Legitimate Script Access Is Blocked

This may occur if:

- a file is intended to be accessed directly
- custom logic depends on direct execution

To resolve:

- adjust the rule
- allow specific files if necessary
- revert the change if required

No Change Observed

- verify rules are applied correctly
- check server configuration
- confirm no conflicting rules exist

Best Practices

- avoid direct access to internal PHP files
- route execution through WordPress where possible
- test all custom functionality after applying
- combine with other execution restrictions

Related

- Prevent Execution in Sensitive Directories
- Restrict Sensitive Endpoints
- 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