File & Execution Protection

File & Execution Protection

- Uploads PHP Execution Protection
- Block Direct PHP Access
- Prevent Execution in Sensitive Directories

Uploads PHP Execution Protection

What This Does

This protection prevents PHP files from executing within the WordPress uploads directory.

It ensures that any uploaded files cannot be used to run code on your server.

Why It Matters

The uploads directory is writable by WordPress.

If an attacker is able to upload a malicious PHP file, they may be able to execute it and gain control of your site.

Blocking PHP execution in this directory removes a common attack vector.

When to Apply It

This protection is recommended for most WordPress sites.

Apply it when:

- your uploads directory is used for media files only
- you do not intentionally run PHP from uploads
- you want to reduce risk from file upload vulnerabilities

When Not to Apply It

Do not apply this protection if:

- your site intentionally executes PHP from the uploads directory
- a plugin or custom functionality depends on this behavior

If unsure, apply cautiously and test your site.

How Steel Security Applies This

Steel Security applies this protection by modifying server behavior.

Depending on your environment, this may include:

- Apache (.htaccess) rules
- Nginx configuration guidance
- IIS web.config rules

These rules prevent PHP files from executing within the uploads directory.

What to Expect After Applying

After applying this protection:

- PHP files in uploads will no longer execute
- media uploads will continue to function normally
- your site becomes more resistant to file upload exploits

How to Verify

To verify the protection:

1. Attempt to access a PHP file within the uploads directory
2. Confirm that execution is blocked

You may see:

- a denied access response
- the file downloaded instead of executed
- an error from the server

How to Revert (Rollback)

To revert this protection:

1. Navigate to the hardening section in Steel Security
2. Disable the control
3. Confirm the change
4. Test your site functionality

The uploads directory will return to its previous behavior.

Common Issues

File Downloads Instead of Executing

This is expected behavior.

Execution is being blocked as intended.

Plugin Stops Working

Some plugins may rely on executing PHP in uploads.

If this occurs:

- revert the change
- investigate plugin behavior
- consider alternative configurations

Related

- Safe Rollback Practices
- Working with Scan Findings

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

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