IIS (web.config) Hardening

What This Covers

This page explains how Steel Security applies hardening using IIS `web.config` configuration.

It provides insight into how server-level protections are implemented for Microsoft IIS environments.

What Is web.config

`web.config` is a configuration file used by Microsoft IIS servers.

It allows you to define rules that control:

- access to files and directories
- request handling
- security behavior

These rules are processed by the server before requests reach WordPress.

Why Server-Level Hardening Matters

Server-level rules provide protection before your application is reached.

This helps:

- block malicious requests early
- reduce load on WordPress
- enforce consistent security behavior

How Steel Security Uses web.config

Steel Security applies hardening by updating the `web.config` file.

These updates may include:

- blocking access to sensitive files
- restricting directory access
- limiting execution in specific locations
- controlling access to endpoints

Changes are applied in a controlled and reversible way.

What to Expect

After applying hardening via `web.config`:

- certain requests may be blocked before reaching WordPress
- restricted files or endpoints will return errors (e.g., 403 Forbidden)
- security rules are enforced consistently

Most changes take effect immediately.

How to Verify

To verify `web.config` hardening:

1. Test access to restricted files or endpoints
2. Confirm that access is denied where expected
3. Observe server responses

You may also review IIS logs if available.

How to Revert (Rollback)

To revert changes:

1. Navigate to the hardening section in Steel Security
2. Disable the relevant control
3. Confirm the change
4. Test affected areas

Steel Security will remove or adjust the rules it applied.

Common Issues

Site Becomes Inaccessible

This may occur if:

- rules are too restrictive
- configuration conflicts exist

To resolve:

- revert the most recent change
- review applied rules
- test incrementally

Changes Do Not Apply

- ensure IIS is reading the updated `web.config`
- check for higher-level configuration overrides
- verify file permissions

Unexpected Behavior

- review recent changes
- test one rule at a time
- check server logs for details

Best Practices

- apply changes incrementally
- test after each change
- keep a backup of your `web.config` file
- avoid manual edits unless necessary
- use Steel Security controls for consistency

When This Applies

This page is relevant if your server uses Microsoft IIS.

If you are unsure:

- check with your hosting provider
- review your server environment
- inspect hosting configuration tools

Related

- Apache (.htaccess) Hardening
- Nginx Hardening
- Applying Hardening Safely
- Safe Rollback Practices

Revision #1
Created 2026-04-04 18:49:19 UTC by Jason Wassing
Updated 2026-04-04 18:49:20 UTC by Jason Wassing