# 10 Nginx rules to enhance WordPress security

When thousands of websites ran on Nginx, some basic tips or Nginx rules were collected to enhance WordPress site security. Take a look at TipsMake.com now!

By far, WordPress is the most popular CMS with over 30% of the web market share. With such a market share, WordPress often becomes a target of security threats. So, for WordPress website owners, it's better to take some measures to tighten the security of your website.

When thousands of websites ran on Nginx, some basic tips or Nginx rules were collected to enhance WordPress site security. Take a look at **TipsMake.com** now!

## Enhance WordPress security with the following 10 Nginx rules

1. 1. Restrict access to XMLRPC
2. 2. Limit the types of requests
3. 3. Access PHP files directly
4. 4. Dotfiles
5. 5. Hide Nginx and PHP versions
6. 6. Security title
7. 7. Block sub-directory access
8. 8. Reduce spam
9. 9. Limit request
10. 10. Disable directory listing

## 1. Restrict access to XMLRPC

XMLRPC endpoints in WordPress are used to allow an external application to interact with WordPress data. For example, it may allow adding, creating or deleting a post. However, XMLRPC is also a common attack vector, where an attacker can perform such activities without permission. It is better to allow XMLRPC requests from an authorized IP that you trust, as follows:

```
location ~* /xmlrpc.php$ { allow 172.0.1.1; deny all; }
```

After adding the code above, you will see **the error code 403** when loading xmlrpc.php in the browser.

## 2. Limit the types of requests

Most of the time the site can only make two types of requests, i.e. **GET** to fetch data from the site and **POST** to upload data to the site. Limiting the type of requests that a site can handle only these two is a good idea.

```
if ($request_method !~ ^(GET|POST)$ ) { return 444; }
```

## 3. Access PHP files directly

If somehow a hacker succeeded in accessing the PHP file into the site, then they would be able to run this file by loading the file, effectively becoming a backdoor to gain access to that website. You should disable direct access to any PHP file by adding the following rules:

```
location ~* /(?:uploads|files|wp-content|wp-includes|akismet)/.*.php$ { deny all
```

## 4. Dotfiles

Similar to PHP files, a dotfile like **.htaccess, .user.ini** and **.git** can contain sensitive information. For added security, it is better to disable direct access to these files.

```
location ~ /.(svn|git)/* { deny all; access_log off; log_not_found off; } locatio
```

## 5. Hide Nginx and PHP versions

Some of the best information should not be disclosed as Nginx version as well as PHP version. This in itself does not prevent the attack. However, assuming a specific version of Ningx or PHP has a flaw, the attacker will not be able to identify it easily from the site. To hide the Nginx version, do the following:

```
#Hide the nginx version. server_tokens off; #Hide the PHP version. fastcgi_hide_
```

## 6. Security title

Security headers provide an extra layer of security by dictating browser behavior. For example, **X-Frame-Options** will prevent the website from loading from an iframe, unless it comes from your own site. **Strict-Transport-Security** will force the browser to load the website from HTTPS.

```
add_header X-Frame-Options SAMEORIGIN; add_header Strict-Transport-Security "max
```

## 7. Block sub-directory access

If the site runs on a subdirectory like **/ blog,** then it is better to allow access to this subdirectory. That means that any shady access to other folders that an attacker is always looking for, for example, **/82jdkj/?.php** is blocked.

```
location ~ ^/(?!(blog)/?) { deny all; access_log off; log_not_found off; }
```

## 8. Reduce spam

Spam comments, while not likely to break your site, will flood the database with junk or malicious content, which can be leveraged as a vector. To reduce spam items, you can add the following rules to your Nginx configuration along with Spam protection plugin like Akismet.

```
set $comment_flagged 0; set $comment_request_method 0; set $comment_request_uri (
```

## 9. Limit request

The WordPress login page, wp-login.php, is the common endpoint for a brute-force attack. An attacker will try to 'take' the site down by sending a combination of usernames and passwords. This is usually done several times a second.

For this, you can apply a rule that limits the number of requests that the page can process per second. Here, the example sets the limit to 2 requests per second. Otherwise, the request will be blocked.

```
limit_req_zone $binary_remote_addr zone=WPRATELIMIT:10m rate=2r/s; location ~ wp
```

## 10. Disable directory listing

Last but not least, you should disable directory listing so that an attacker is not aware of what is in the directory. There are very few reasons that directory listing is useful on a WordPress site.

```
autoindex off;
```

You finished reading the article "**10 Nginx rules to enhance WordPress security**" edited by the **TipsMake** team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.