SiteGround

SECURITY MATTERS:
21 Tips to Keep Your WordPress Secure

INTRODUCTION

INTRO

Wondering if WordPress is secure? The answer is yes. WordPress is built with the latest security technology and has a regular release schedule that includes up-to-date vulnerability patches. It's also monitored and maintained by a large community of developers who care deeply about security.

If WordPress is so secure, why did we prepare a guide on preventing hacks? The truth is, protecting your site is an ongoing practice. It involves building barriers for attackers, preventing failures, tracking changes, denying malicious access, hiding sensitive information, and more.

WordPress powers more than 30% of websites and its popularity makes it a target for cyber-criminals. It's also a platform in continuous development, open to changes and integrations with third parties.

At SiteGround, we are aware of the many ways your site can fall prey to attacks and we're committed to helping site owners protect themselves. Use this guide to increase your knowledge of WordPress web security, implement new measures to secure your website, and spread the word with those around you about the need to protect their sites.

1

PROTECT YOUR FILES AND DATABASES

1. BEFORE INSTALLING WORDPRESS

When setting up a new WordPress installation, you should always choose the latest stable version. Before installing it, follow these two simple web security steps in the wp-config.php file:

- *Change the database prefix*
- *Use authentication keys*

By default, all WordPress installations use the prefix **wp_** for their database. This is consistent throughout WordPress, so it's recommended to change the prefix for each site to prevent possible attacks related to the database.

To alter the WordPress table prefix, change the following line in the configuration file, wp-config.php, with the prefix that you would like to use:

```
$table_prefix  = 'wp_';
```

For example:

```
$table_prefix  = 'newsite_wp_';
```

This change will also allow you to have several WordPress installations on the same database, as long as you do not repeat the prefix.

If your site is already installed and you didn't change the default prefix during the installation process, it's not too late. Use a plugin such as Change Table Prefix to make modifications. You can also do this manually, but I don't recommended it if you are not familiar with performing database changes.

WordPress has secret keys, called Keys and Salt, that are stored in the wp-config.php file. They protect open sessions by encrypting the session data in the browser's cookie. Before beginning the installation, you should generate the secret keys.

Like with the database prefix, you can change the secret keys on an existing site, at any time, a task that I recommend you perform routinely in order to invalidate active sessions and force all users to log in again.

Although you can generate your own keys manually, I recommend using the official WordPress service found at **https://api.wordpress.org/secret-key/1.1/salt/** and replace the keys with the ones in your wp-config.php file.

Before moving on to the next tip, I have one more piece of advice about WordPress secret keys for live sites. In the unlikely scenario where you need to deny any type of access to the admin panel, even with login credentials, you can configure keys to invalidate every microsecond by replacing them in the wp-config.php with the following:

```
define('AUTH_KEY',         microtime());Lorem
define('AUTH_KEY',         microtime());Lorem
define('AUTH_KEY',         microtime());Lorem
define('AUTH_KEY',         microtime());Lorem
define('AUTH_KEY',         microtime());Lorem
define('AUTH_KEY',         microtime());Lorem
define('AUTH_KEY',         microtime());Lorem
define('AUTH_KEY',         microtime());Lorem
```

Remember to routinely update these keys as a preventative measure or to end active sessions.

2. AFTER INSTALLING WORDPRESS

Once you've finished installing your new WordPress site, you should delete the admin profile used during installation and create a new user with admin permissions, as well as any other necessary user accounts.

Avoid weak usernames like admin or administrator which are common in all WordPress installations and remember to use a strong password.

Disable the pingbacks and trackbacks notifications on your admin panel (Settings > Comments), as they can be an entry for possible DDoS (Distributed Denial of Service) attacks on your site.

Protect files from attacks and intrusions by adding the following lines of code in the htaccess file. Ideally, this should be done at the beginning of the file located at the root directory of your site:

```
#Deny Directory Listing
Options - Indexes
#Block sensitive files
<files.htaccess>
Order allow,deny
Deny from all
</files>
<files wp-config.php>
Order allow,deny
Deny from all
</files>
```

You should block access to any unnecessary files by creating a new .htaccess file in the /wp-admin directory and adding the following lines of code:

```
#Block installation files
<files install.php>
Order allow,deny
Deny from all
</files>
<files setup-config.php>
Order allow,deny
Deny from all
</files>
```

I recommend reviewing the robots.txt file, which is located in the root directory of your site. This file tells search bots what should and shouldn't be analyzed on your site, so be sure to check that it doesn't show any sensitive information about your WordPress installation, for example, your wp-admin folder.

3. CHANGE PERMISSIONS FOR FILES AND DIRECTORIES

Make sure files and directories in your WordPress installation have the appropriate permissions to prevent attackers from taking control of your site.

You can change the permissions through an FTP client or through an admin panel provided by your web host. With SiteGround, it's easy to change file and folder permissions in your cPanel.

Go to WordPress Tools > WordPress Toolkit > Select the installation > Fix Permissions.

- *Permissions for all the directories should be set to 755.*
- *Permissions for all the files should be set to 644.*

To restrict the access even further, you should protect these two files in your WordPress configuration in the following way:

- *wp-config.php file: set permissions to 600*
- *.htaccess file: set permissions to 604*

These permissions are referred to as View, Write, and Execute as defined in Unix operating systems.

4. BLOCK PHP IN DIRECTORIES

Although WordPress installations block PHP file uploads through the admin panel by default, you should block the option to execute PHP code in that folder. You should also limit the unnecessary execution of PHP code in other folders used by WordPress that shouldn't be accessed directly.

Create a new .htaccess file inside of your folders "/wp-content/uploads", "/wp-content/plugins" and "/wp-content/themes", and add the following lines of code to block PHP executions:

```
<Files *.php>
deny from all
</Files>
```

Note: take into account that after every modification in a .htaccess file, you should check it in your installation. Flush the cache to confirm that the added rules are working properly.

5. DISABLE FILE EDITING IN WORDPRESS

This step focuses on adding a layer of security to the admin panel to prevent unwanted intruders and limit mistakes made by authorized users.

To disable the file edit option in the WordPress admin panel, use the following line of code in the configuration file, wp-config.php:

```
define( 'DISALLOW_FILE_MODS', true );
```

This code is equivalent to removing the 'edit_themes', 'edit_plugins', and 'edit_files' permissions for any registered user on the site.

You can add an additional layer of control for live sites if you don't want users to install themes and plugins on their own. To do this, add the following code to the configuration file - wp-config.php:

```
define( 'DISALLOW_FILE_MODS', true );
```

Remember to deactivate it by changing the directive to false if you need to perform tasks on the WordPress installation.

All modifications on the wp-config.php file above the following line of code:

```
/* That's all, stop editing! Happy blogging. */
```

6. USE A CDN AS A DNS

Although we already know the benefits of a Content Delivery Network (CDN) service to improve your website performance, using a DNS-type CDN (before your web server) can improve your web security in the following three ways.

- *It enables an active Firewall that is updated continuously against malicious behaviour like massive connections, tracking ports, etc.*
- *It prevents brute force attacks by using the distributed server network of the provider which minimizes the impact and applies blocking rules to detect these kinds of attacks, usually DoS or DDoS.*
- *It hides the real IP of your server which prevents direct attacks against your site by masking the real IP where you are hosted.*

I recommend using CloudFlare as your CDN to improve the security and performance of your WordPress website. All SiteGround hosting plans include a free CloudFlare account.

7. BACKUP YOUR SITE

Although we hope you'll never have to use this tip, it's better to be safe than sorry and have a full backup of your site.

You rarely need to restore a full site backup, but in case you do, SiteGround has a tool for backups and easy restores developed inhouse, independent of the web service infrastructure. You can rest easy knowing we have copies of your files in case of any incident, and you'll be able to restore your site easily and quickly.

I recommend you follow the 3-2-1 rule as a strategy for backups that contain important data.

- *Keep 3 backups*
- *In 2 different formats (minimum)*
- *1 of the backups should be in a different physical location*

In case disaster strikes, it's useless to have all your backups in the same format or location. Remember to always generate a new backup after you make any important changes to your WordPress installation.

2

SECURE YOUR LOGIN AND SESSIONS

8. ACTIVATE AND FORCE HTTPS

The HTTPS protocol creates a secure connection between users and the server, eliminating possible **Man-in-the-Middle (MITM)** attacks. These attacks happen when an intermediate service alters or acquires information exchanged between two ends. That's why we use HTTPS encryption for all sensitive information.

To use the HTTPS protocol on your site, install an SSL certificate on your web server and change the URL in the admin panel.

With SiteGround, all hosting plans include free Let's Encrypt SSL certificates that can be installed and configured with an easy tool in the control panel under Security section > Let's Encrypt.

There are several WordPress plugins that force an HTTPS connection on all your site resources, to avoid warnings or errors when serving both HTTP and HTTPS content on the same page.

Finally, you must force any new session in the admin panel of your site to be under SSL protocol by adding the following code to the wp-config.php file:

```
define('FORCE_SSL_LOGIN', true);
define('FORCE_SSL_ADMIN', true);
```

Note: remember that you must have an active SSL in your installation, for example, the one provided by Let's Encrypt.

9. DISABLE SESSION SUGGESTIONS

As previously mentioned but worth repeating, giving as little information as possible to attackers should be your first priority. This tip will help you minimize possible entries to your site by disabling the login suggestions from the login page, which appear by default if the username or the password are incorrect.

```
function no_wordpress_login_errors(){
return 'Thanks for trying, but we are
}
add_filter( 'login_errors', no_wordpress_login_errors );
```

Note: you can customize the message.

10. MOVE YOUR SITE'S ADMIN ACCESS

It's no surprise that many website attacks happen on the login page. That's because bots are programmed to recognize a WordPress installation and add the path /wp-admin to get to the login page. Then, they can easily force entry if the usernames and passwords are weak. All WordPress sites use the same path for the login page, which means changing it will add another layer of difficulty for attackers.

There are several plugins in the WordPress repository that allow you to change the path and location of your login page, for example, **www.mydomain.com/newadminpanel**

I recommend the WPS Hide Login plugin. However, other plugins exist and many security plugins also include this functionality.

11. LIMIT LOGIN ATTEMPTS

You can configure your site to block access to the login page for a few minutes, a few hours, or permanently when a user inputs incorrect login credentials a certain number of times. This makes it more difficult for bots to gain access through brute force attacks.

Security plugins like Wordfence normally include this feature, as do the following plugins:

- *Limit Login Attempts (miniorange)*
- *Limit Login Attempts Reloaded*
- *Loginizer*

Some firewall plugins also include this functionality.

12. USE FIREWALL PLUGINS

A Firewall is an additional layer of security software that can protect your web connections or your WordPress installation by detecting and analyzing incoming connections. Firewall plugins are very effective and easy to manage, since everything is configured from a single plugin.

They normally include a Firewall WAF (Web Application Firewall), a tool that analyzes and blocks attacks to the website in real time. At SiteGround, our customers have this service by default. We analyze the types of connections and block attack attempts in a completely transparent way for our customers.

Some of these plugins are:

- *Wordfence Security (be careful with the Live Traffic functionality, which may leave you without service due to server overload)*
- *All in one security and firewall*
- *iThemes Security (previously Better WP Security)*

Some firewall plugins also include these functionalities:

- *File scanner to search for changes, errors, and viruses*
- *Firewall WAF that detects and blocks malicious visits*
- *Real-time traffic viewer*
- *A tool to block access to the website by IP*
- *Captcha for the WordPress login page and a limit login attempts feature*
- *Password audit*

- *Two-Factor Authentication to access the WordPress admin panel*
- *Ability to block specific countries*
- *A tool to check folder and file permissions*

13. USE SECURITY HEADERS

Improve your website security by implementing a series of headers incorporated into the web server and sent to the browser.

Start with the X-Frame-Options header, which prevents pages from being opened in an external frame or iframe, which prevents clickjacking attacks on your website: a technique that tricks internet users into revealing confidential information on a seemingly normal website.

By adding the following line of code to your .htaccess file, you tell the browser that frames can be only opened from the same domain or origin:

```
Header set X-Frame-Options SAMEORIGIN
```

If your website includes services that can be embedded by third parties, you can specify which domains are allowed and deny access the rest. For example:

```
Header set X-Frame-Options "ALLOW-FROM https://example.com/"
```

Increase your site's protection against XSS (cross-site scripting) attacks on older browsers by adding the following line of code to your .htaccess file:

```
Header set X-XSS-Protection "1; mode=block"
```

To reduce the risk of XSS, build on the following tip by using the content-security-policy header or browser content security policy, which specifies what content from your site or third parties is allowed to dynamically load.

For example, if you want your site to only accept content from the same domain, add the following line of code to your .htaccess file:

```
Header set Content-Security-Policy "default-src 'self';"
```

This blocks scripts from loading from external sources.

To modify the variables for your specific project, for example, to allow scripts from third parties like Google Analytics, use this line of code:

```
header set Content-Security-Policy "script-src 'self'
www.google-analytics.com;"
```

This header should be carefully implemented because it's easy to block resources without noticing. I recommend performing different tests with this header in a separate browser tab to check for errors on the terminal.
Note: if you previously included the x-content-security-policy header in your server and it's outdated, you'll need to delete it as it might cause issues if you use both headers at the same time.

The fourth header you can use to boost your security is the X-content-type-options, which protects you from unwanted styles and scripts to load when the expected MIME types do not match what was declared on the page. To add this protection, add this line to your .htaccess file:

```
Header set X-Content-Type-Options "nosniff"
```

14. PREVENT XML-RPC ATTACKS

The xmlrpc.php file is used by some applications and softwares to communicate with WordPress, for example the WordPress app or mail clients like Outlook and Thunderbird that allow the feature "publish by email". Plugins like Jetpack or the Json Api use the XMLRPC file for some of their functions.

You can completely deny access to the xmlrpc.php file by using rules in the .htaccess file or by deleting it if you are sure that you don't need it.

To deny access via .htaccess, add the following lines of code to the file:

```
# deny access to xmlrpc.php
<Files xmlrpc.php>
Order Deny,Allow
Deny from all
</Files>
```

You can also use plugins like Disable XML-RPC or iThemes Security, mentioned in tip #14, to deny access to XMLRPC.

For those who absolutely need this API functionality, the best solution is to enable it only from the IP where you need access and deny the rest. In this specific case, add the following lines of code to the .htaccess file, modifying the IP to the one requiring access:

```
<Files xmlrpc.php>
 order deny, allow
 deny from all
 allow from X.X.X.X
</Files>
```

15. DISABLE JSON REST API

Since WordPress version 4.4, the REST API is included in the core software, allowing any developer to interact with the site. This allowed WordPress to reach a greater number of developers not familiar with WordPress, but at the same time, leaves an open door to possible attacks to your site, especially DDoS attacks.

If none of your plugins use the REST API, you can easily deactivate it for your installation. Simply add the following lines of code to the functions.php file of your active theme or resources plugin:

```
add_filter('json_enabled', '__return_false');
add_filter('json_jsonp_enabled', '__return_false');
```

If you would rather not tinker with the code, you can use the plugin Disable REST API. You can also use the iThemes Security plugin, mentioned in tip #12 about Firewall Plugins, that will keep the REST API active but allow access only to users with exclusive permissions.

3

MAINTAIN A SECURE WORDPRESS INSTALLATION

16. CHOOSE REPUTABLE PLUGINS AND THEMES

Plugins and themes are powerful third-party resources that can help you increase the functionality of your WordPress site. There are hundreds of thousands of them available both in the official WordPress repository and elsewhere online. Since not all of them are vetted, this represents a serious security problem. Most of us don't carry out exhaustive code and functionality reviews before installing a plugin but a questionable plugin can cause security breaches and conflicts.

Only download plugins and themes from the WordPress repository and reputable sites. Before choosing your next plugin or theme, I recommend you:

- *Take a look at the reviews, number of downloads, and comments.*
- *Look at the latest update to see if the software is actively maintained.*
- *Research the author and look at any other themes they've built in the repository.*
- *Check for compatibility issues with the software and your current installation.*

Always perform a full website backup before installing a new plugin or theme.

17. DELETE WORDPRESS VERSION INFORMATION

This security tip will hide information about your WordPress version from the HTML code on your site. This will prevent attackers from capitalizing on any known vulnerabilities associated to a particular version of WordPress.

You can delete the information from the HTML header and from the static files by adding the following code to the functions.php file of your theme or in the utilities of your plugin:

```
/*
Hide scripts and style version
*/
function SG_remove_wp_version_strings( $src ) {
        global $wp_version;
        parse_str(parse_url($src, PHP_URL_QUERY), $query);
        if ( !empty($query['ver']) && $query['ver'] === $wp_version ) {
                $src = remove_query_arg('ver', $src);
        }
        return $src;
}
add_filter( 'script_loader_src', 'SG_remove_wp_version_strings' );
add_filter( 'style_loader_src', 'SG_remove_wp_version_strings' );
/*
Hide generator tag from the header
*/
function SG_remove_wp_generator() {
        return '';
}
add_filter('the_generator', 'SG_remove_wp_generator');
```

You can also hide information about the current WordPress version by adding the following line of code to the .htaccess file inside the WordPress root directory:

```
#Block WP info
<files readme.html>
Order allow,deny
Deny from all
</Files>
<files license.txt>
Order allow,deny
Deny from all
</files>
```

Note: although some WordPress security guides recommend deleting these files altogether, my recommendation is to block their access since a new WordPress update or re-installation might generate a new one.

18. HIDE PHP WARNINGS AND NOTICES

Along with other strategies to limit information given to attackers, it's a good idea to hide error reports. Error reports can provide valuable information to attackers such as your site's PHP and WordPress versions, the path to your folders, or server information.

In development environments, error reports are useful to validate your work and find potential mistakes, however, on a live site, you should deactivate these records to hide information such as paths, names, versions, and more.

To disable error reports in WordPress, simply add the following lines of code to the wp-config.php file:

```
error_reporting( 0 );
ini_set( 'display_errors', 0 );
```

19. HIDE APACHE AND PHP INFORMATION

The last tip on hiding information is to configure the headers sent by the servers. These often contain information about the software installed on the server and the PHP version being executed.

Depending on the installation, you should hide or limit the information being shared about the web server by adding the following line of code to the .htaccess file in the root directory:

```
ServerSignature Off
```

There are two ways to hide the information about your site's PHP version that some servers send in the HTTP header. First, add the following code to the .htaccess file:

```
Header unset X-Powered-By
```

Or use the following directive in the php.ini:

```
ini_set( 'dispexpose_php = Offlay_errors', 0 );
```

Note: normally you can add this line of code to your active php.ini through the server admin panel, but this may be different depending on your hosting provider.

20. KEEP YOUR WORDPRESS UPDATED

To protect your website against known security vulnerabilities, you should use the latest version of the WordPress core software, keep any installed plugins updated, and update your themes.

I personally prefer to maintain my site manually, although it requires more attention and time because it allows me to be aware of the features included in every update and the reason for them. In terms of the update order, I always recommend updating the WordPress core software first and then, without any specific order, updating plugins and themes.

If, however, you want the WordPress core to be updated automatically, simply add the following line of code to your wp-config.php file:

```
define( 'WP_AUTO_UPDATE_CORE', true );
```

Note: the automatic update won't run if you have disabled the WordPress cron.

You'll receive an email to the address used by the platform admin account after every update.

Updating WordPress core is only one part of the equation. Based on a report from wpscan.org, 52% of vulnerabilities found in WordPress installations are due to plugins, 11% due to themes, and 37% due to the WordPress core software.

If you want to update plugins automatically, add the following line of code to the functions.php file of your active theme or in your functionality plugin:

```
add_filter( 'auto_update_plugin', '__return_true' );
```

Before adding this code, delete all the unused plugins on your site. Simply deactivated them is not enough to remove potential vulnerabilities. Delete them!

Adding the following line of code will automatically update themes:

```
add_filter( 'auto_update_theme', '__return_true' );
```

Lastly, remember that keeping your WordPress site secure is great, but the computer you use should also be protected from malicious software and viruses. Make sure you use a reputable antivirus and your operating system is up to date.

21. CHOOSE A TRUSTED HOSTING COMPANY

The last security tip, although it should be the first in terms of importance, is to choose a secure server to host your web project.

Your hosting provider should offer you a secure platform and actively maintain the security of their infrastructure. Be wary if your host uses outdated software, unsecured access, and if their tech support has little knowledge of WordPress.

Choosing the correct hosting provider will influence the success and security of your WordPress project in a big way.

CONCLUSION

RESUME

They say common sense isn't so common, but in terms of security, it's your best ally. Use strong passwords, delete inactive users, assign the right roles to each user, do not save active sessions on public computers, keep the server time updated, only allow secure access, and actively monitor your website.

Of course, there is no such thing as a 100% secure website, but I hope this guide can help you maintain a safe site for a long time. Use your judgment when applying these recommendations and only use the tips you need and which are compatible with your project. Staying on top of the latest news and trends in WordPress and web security can help you as well.

This guide is based on my experience of over 10 years working with WordPress. The knowledge collected and shared in this guide exists largely because of the professionals in the community to whom I am very grateful. These tips were also sourced from a multitude of online resources on WordPress web security, content which is being generated almost daily, so, there's a good chance by the time you read this guide, some tips will already be outdated and others will not be necessary due to WordPress core updates.