Security Hardening WordPress

WordPress security is as good as you make it and you can always make it better. This post will discuss tools and techniques for making your WordPress installation more secure.

Some of these are default choices in WordPress but some are not only not the defaults but you have to manually added to configuration files or .htaccess files.

Some of these rules will make local development harder as we expect constant change. However, these are excellent for production sites where we don't expect major changes and we expect them to be slower in adapting new code.

> This post links to third party plugins in the WordPress Plugin Directory. They are what I use and they've proven good for my use.
>
> Please, always test third party plugins and themes on a development environment away from production data and users.

Disable automatic registration

The first idea is an easy one. Disable automatic user account creation for your site.

Yes, I get that we're disallowing legitimate users from logging in to the site but I consider this a fair price to pay to prevent bots and other bad actors from running crazy on my sites.

It happens that this is the easiest task to complete. From Inside the Administrator Interface go to ***General*** and about midway down the screen you will see option to allow anyone to register and a pulldown menu to indicate the new user's default role.

Figure 1: WordPress settings for

membership
and new
user default
role.

Make sure that the default values shown in the figure above remains as they are unless you're ready to do the work of managing users and their roles.

Unless you have a lot of users you can create accounts and assign roles manually.

Check file permissions

The next item on my list is to check the permissions of the files in my WordPress installation. If you are used to the octal three-digit representation of Unix permissions we want our directories to have permissions of 755 and our files to have permission of 640.

We don't want anyone but the owner of the files, the user who WordPress was installed under, to have access to the WordPress files.

Protect configurationn file from direct access

Next on the list, we need to make sure that our configuration file, `wp-config.php` is not readable by anyone who doesn't need to. The configuration file containes the name and credentials for the WordPress database and the salt for all our security tools like nonces.

To disable access we add the following code to the `.htaccess` file at the root of our WordPress installation.

Make sure this code is at the top of the file, before the generated code for permalinks.

```
<Files wp-config.php>

    # Apache < 2.3
    <IfModule !mod_authz_core.c>
```

```
        Order allow,deny
        Deny from all
        Satisfy All
    </IfModule>

    # Apache >= 2.3
    <IfModule mod_authz_core.c>
        Require all denied
    </IfModule>

</Files>
```

Use .htaccess to prevent access to files

In addition to making sure people can snoop on our configuration file we need to make sure that they can't guess what files we have stored in our WordPress installation.

It uses Apache's mod_alias and the RedirectMatch directive to redirect a list of indicated files and extensions to a 403 status code, denying access to the requested resource.

```
# SECURE LOOSE FILES
<IfModule mod_alias.c>
    RedirectMatch 403 (?i)(^#.*#|~)$
    RedirectMatch 403 (?i)/readme\.(html|txt|md)
    RedirectMatch 403 (?i)\.(ds_store|well-known)
    RedirectMatch 403 (?i)/wp-config-sample\.php
    RedirectMatch 403 (?i)\.(7z|bak|bz2|com|conf|dist|fla|git|inc|ini|log
</IfModule>
```

With these rules we make sure that people can't get our backups and other sensitive files directly from the server.

Prevent administrators from editing theme and plugin files

Depending on the situation you may not want administrators to edit themes and plugins from the admin UI. This is particularly important when contractors and other third parties have access to admininsrative areas and, if allowed, can edit any plugin or theme installed on the server and make unexpected changes and introduce malware to your system.

To disable editing, add the following command to your `wp-config.php` if it doesn't exist already.

To Reenable editing set the attribute to false or remove it altogether.

```
<?php
define('DISALLOW_FILE_EDIT', true),
```

Disable Directory Listing

Another way for external users to snoop in and potentially gain information about your system to use in an exploit is to see what files are on your WordPress installation by being able to see the files inside your installation.

The easiest way to fix this issue is to add an empty `index.html` or `index.php` in directories where one is not available

Remove Version Number

Surprisingly the version number of WordPress can be used as a weapon by hackers and other bad actors. Once they identify a vulnerability in a given version of a software they will look for it so they can use the vulnerabilities in those specific versions.

The code goes in the theme's `functions.php` and it does the following:

1. Removes the version number in HTML pages by removing the `generator` meta tag

2. Removed the version number from the RSS feed by returning an empty string instead of the generator string
3. Removes the version query string attribute from scripts
4. Removes the version query string attribute from style sheets

```
<?php
remove_action('wp_h'wp_head'_generator'); // 1

add_filter('the_generato'); // 1

add_filter('ring'); // 2

// remove version from scripts and styles
function rivendellweb_remove_version_scripts_sty'); // 2

// remove version from scripts and styles
function rivendellweb_remove_version_scripts_styles($src) {
    if (strpos($src, '',
  'rivendellweb_remove_version_scripts_styles', 9999); ',
  '// 3ilter(
  'script_loader_src',
  'rivendellweb_remove_version_scripts_styles', 9999); // 4
'script_loader_src'// 4
```

These changes don't prevent WordPress from working but it will remove the version of WordPress from all the places it appears in the WordPress pages and feeds.

Limit Number of Login Attempts

One basic way to reduce the impact of bad actors getting user names for a brute force attack is to limit the number of times they can attempt to log into your system

Cerber Security, Antispam & Malware Scan allows you to control login attempts and disallows further access when the limits are exceeded.

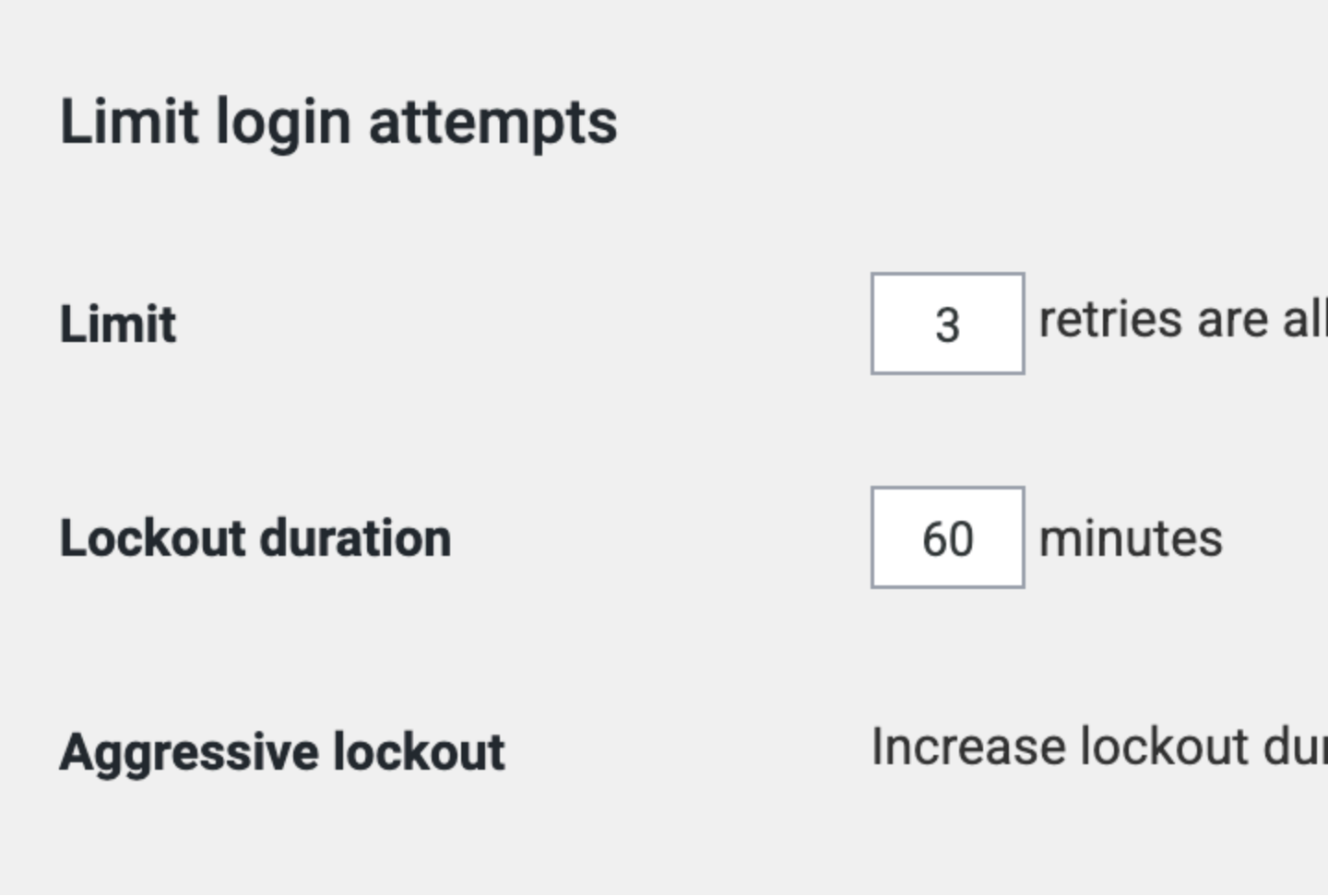

Figure 2: Cerber Security Plugin Login Configuration

Stop User Enumeration

User Enumeration is a type of attack where nefarious parties can probe your website to discover your login name. This is often a pre-cursor to brute-force password attacks.

The idea is to block the user enumeration attack and make the following brute force attack harder.

For example, a hacker trying to attack your site could run the following URLs to hit the REST API in order to get information about users in our WordPress installation.

```
http://example.com/wp-json/wp/v2/users/1
http://example.com/wp-json/wp/v2/users/2
http://example.com/wp-json/wp/v2/users/3
```

```
http://example.com/wp-json/wp/v2/users/4
http://example.com/wp-json/wp/v2/users/5
```

We could also automate the discovery with something like CURL or Wget.

There purpose built plugins against these types of attacks like Stop user enumeration

General purpose security plugins like Cerber Security, Antispam & Malware Scan also have the option to stop enumeration attacks.

Monitor Usage

This is important if you use contractors as part of your admin team and you want to keep track of what's going on in your syste.

Use a pluging like Activity Log to track activity on your server. This answers the "who did it" question

If you haven't disabled editing access for administrators then a plugin like Website File Changes Monitor will let you see what files were added, modified or deleted from your WordPress installation. THis answers the "what was done" question.

Things To Consider

These are not hard requirements but things that are worth considering and that may improve security for your users but it takes work from them.

Do You Want to Allow Hotlinking?

Hotlinking is when websites other than your own link to your assets. This can have adverse effect on your bandwidth and your content can be used for things you won't aprove of.

To prevent hotlinking you can use the following `.htaccess` fragment.

```
 # STOP HOTLINKING
<IfModule mod_rewrite.c>
  # 1
    RewriteCond %{HTTP_REFERER} !^$
    # 2
  RewriteCond %{HTTP_REFERER} !^http(s)?://([^.]+\.)?rivendellweb\.net [NC
    # 3
  RewriteRule \.(gif|jpe?g?|png|webp|woff2|woff)$ - [NC,F,L]
</IfModule>
```

The idea behind the fragment is that:

1. If there is a referer, meaning the request comes from a fully qualified URL
2. And the URL doesn't match our domain our domain, in this case `rivendellweb.net` either in http or https
3. Then fail all the requests for images and fonts identified by their extension

We can add additional extensions to the list in step 3 to make sure we keep our content on our own server.

Consider Two-Factor Authentication

Two-factor authentication uses a second means of authenticating you before you're allowed access to WordPress.

I ran this two-factor authentication plugin search on the WordPress.org plugin directory. I will not advocate any particular solution.

However if you use Duo Two-Factor Authentication or the Google Authenticator Plugin for other two-factor authentication needss, you can leverage those tools with your WordPress installation.

Two Factor Authentication presents an interesting issue. Unless you make two factor authentication mandatory for all users (which would make sense) you'll have to use something like the Google Authenticator – Per User Prompt oor its equivalent for your authenticator so it will only show it to the users who have opted in to authentication.

Consider A WordPress-Based Firewall

The last consideration we'll adress in this post is whether to add a software firewall in addition to whatever hardware or software firewall your host provides. There are difference between software and hardware firewalls that should be kept in mind when making the decision.

I ran a firewall plugin query on he WordPress plugin directory for reference.

The plugin that caught my attention and I've used in the past is Block Bad Queries (BBQ). The only drawback is that you can't customize the settings, for that you need a pro version