WordPress Security Clampdown

(War Room Edition 4.0)

By

Shaun Pearce

Disclaimer

This report is for information purposes only and is based on the author's own experiences and research.

Use this information at your own risk. The author accepts no responsibility or liability whatsoever for any losses (including but not limited to financial or data losses) that may be incurred as a result of implementing any of the instructions in this report.

The information in this report is accurate as of the date of publication; however, due to the rapidly evolving nature of the Internet some or all of the tactics or plug-ins mentioned in this report may subsequently have changed or be unavailable.

Copyright Info

This report is EXCLUSIVELY for members of the War Room on the Warrior Forum.

YOU **DO NOT** HAVE PLR, RESALE OR GIVEAWAY RIGHTS TO THIS REPORT.

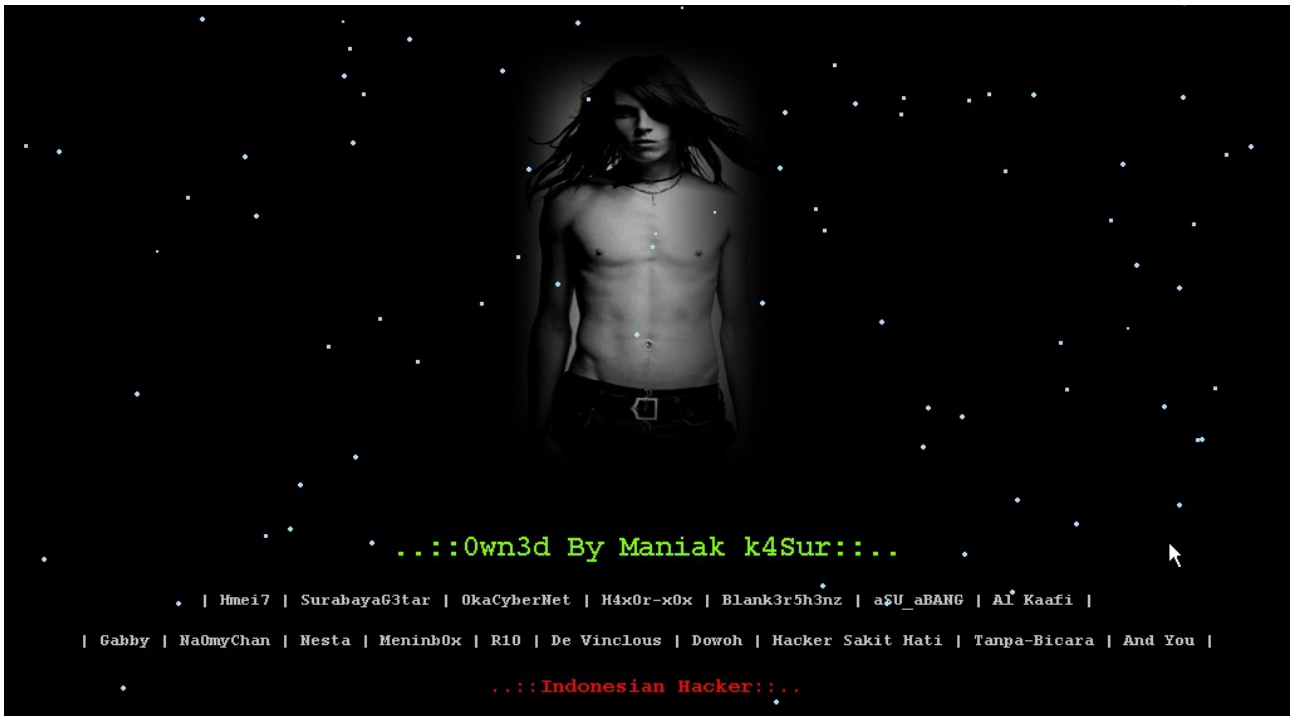
Contents

Table of Contents

I Got Hacked

I had a shock one day when I went to log into one of my WordPress sites. The display kept saying “Incorrect Username or Password”. This was puzzling, as I hadn't changed either, and had not had any trouble logging in before.

I went to the main site, and the reason why I was unable to log in became readily apparent: My site had been hacked!



A garish image, fake snowfall, and triumphant message informed me my site had been hacked by an Indonesian hacker. Just to rub the salt in, a blaring heavy metal soundtrack played, and I was unable to go back to the log-in page without closing my Internet browser and starting all over again. Needless to say, I was very unhappy!

But it got worse...

I had four other WordPress sites on that server. I checked each one in turn and, sure enough, every one of them had been hacked – by the same hacker.

Then I started to get worried. A few years ago, my most lucrative sales site (a simple HTML job) had got hacked. Google flagged it as an attack site, and it took ages to get it sorted out. I lost thousands in sales, and its position in the search ranking tanked.

Fortunately that hadn't happened this time, but the hacker was bragging that he “owned” my sites on hacking forums. Cheeky little b*****!

I had to figure out how to get my sites back PDQ. In the process, I had to

learn about WordPress' vulnerabilities (and boy does it have a few) and try to think like a hacker. Not only did I have to figure out how I was going to get my sites back from the hackers, but I also had to learn how to make my sites unattractive to hackers, so they wouldn't come back.

Touch wood, I've been successful so far, and I'm going to share what I learned with you in this report.

I'll tell you:

- All about WordPress' vulnerabilities – it's crucial you understand these, because hackers surely do.
- How to get your site back after it's been hacked. These are the exact same steps I followed.
- Finally, I'll share the plug-ins I found to protect my sites from hackers. They're free, and they work. There are a lot of them out there, but I'll show you the combination of plug-ins that will give you the best results.

WordPress Vulnerabilities

WordPress is great. It's evolved from being a clunky, blogging platform into pretty much the de facto format for new websites. It's great strength is its open source versatility, but that strength is also a weakness. Being open source, unencrypted .php code, anyone can tear away the surface and look at what's beneath. Any possible loopholes can be laid bare – if you know what you're looking for.

Most people aren't programmers, and don't know what to look for. We (and I include myself in this category) just take WordPress at its face value, and put it to work in the same way as we put our computers to work. We don't need to know how a CPU works, we just take it for granted that it does.

A lot of people are programmers though, and WordPress with its simple, open source architecture is perfect for them to examine, access and improve. Some programmers put this knowledge to good use and report loopholes, bugs and vulnerabilities to WordPress. If you find one yourself, you can report it here: https://make.wordpress.org/core/handbook/testing/reporting-security-vulnerabilities/

Others exploit this knowledge for nefarious purposes, and use it to hack into WordPress sites.

Why do hackers hack?

There are as many answers to that question as there are hackers. Often hackers will want to plant hidden back-links to try and increase their site's position in the search engines' rankings. Others will want to hijack your server to run bots or rogue .php scripts and attack other sites. Others hack just for the heck of it.

The good people who run WordPress try to stay one step ahead of the bad guys, and update WordPress regularly to eliminate known security bugs – which is why you should always update WordPress whenever a new version is released.

It's a hassle because new versions and updates seem to be coming out every few weeks, but it's also imperative because each new update is safer and more secure than the version it replaces. Updating doesn't take up much time and is very easy. All you have to do is click the button, and WordPress does the rest.

There are, however, two core parts of WordPress which are highly vulnerable because they are at the heart of the way WordPress works.

The first one of these vulnerabilities is the WordPress admin area.

Anyone who knows anything about WordPress knows the log-in area can be accessed by going to http://[your URL]/wp-admin/ . It's a major weakness in the system.

When they get there, all they have to do is figure out what the username and password is and, hey presto!, they're into the site dashboard.

The hacker's job is made all the easier by the fact that WordPress sets the username as "admin" by default. Most people don't change this – or if they do, they use easy to guess user-names like "Administrator" or "Webmaster".

Hackers can also work out a username from looking at the posts on a blog. By default, WordPress uses the same name for both the username and display name. If a hacker sees a large number of posts by "admin" or "Fred" or "Jane" chances are one of those will be a username with administrator privileges.

Once the hacker has figured out a user name, they can then start what's known as a "brute force attack". This is where hackers use software or bots to crack a WordPress password.

Wikipedia defines a brute force attack like this:

> *n cryptography, a **brute-force attack**, or **exhaustive key search**, is a cryptanalytic attack that can, in theory, be used against any encrypted data[1] (except for data encrypted in an information-theoretically secure manner). Such an attack might be utilized when it is not possible to take advantage of other weaknesses in an encryption system (if any exist) that would make the task easier. It consists of systematically checking all possible keys or passwords until the correct one is found. In the worst case, this would involve traversing the entire search space.*
>
> *When password guessing, this method is very fast when used to check all short passwords, but for longer passwords other methods such as the dictionary attack are used because of the time a brute-force search takes.*

The longer and more complicated you can make your password, the better.

The biggest vulnerability in WordPress is a file.

That file is called wp-config.php and it's found in the root directory. This is what it looks like:

```
<?pp
/**
 * The base configurations of the WordPress.
 *
 * This file has the following configurations: MySQL settings, Table Prefix,
 * Secret Keys, WordPress Language, and ABSPATH. You can find more information
 * by visiting {@link http://codex.wordpress.org/Editing_wp-config.php Editing
```

```
 * wp-config.php} Codex page. You can get the MySQL settings from your web host.
 *
 * This file is used by the wp-config.php creation script during the
 * installation. You don't have to use the web site, you can just copy this file
 * to "wp-config.php" and fill in the values.
 *
 * @package WordPress
 */

// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define('DB_NAME', 'samowen_wp344');

/** MySQL database username */
define('DB_USER', 'samowen_wp344');

/** MySQL database password */
define('DB_PASSWORD', '64rP6asS4p');

/** MySQL hostname */
define('DB_HOST', 'localhost');

/** Database Charset to use in creating database tables. */
define('DB_CHARSET', 'utf8');

/** The Database Collate type. Don't change this if in doubt. */
define('DB_COLLATE', '');

/**#@+
 * Authentication Unique Keys and Salts.
 *
 * Change these to different unique phrases!
 * You can generate these using the {@link https://api.wordpress.org/secret-
key/1.1/salt/ WordPress.org secret-key service}
 * You can change these at any point in time to invalidate all existing cookies.
This will force all users to have to log in again.
 *
 * @since 2.6.0
 */
define('AUTH_KEY',
'adahh9mcsm6o5mbhro5c9iipbbdcy14n0l6atnv4h53zpgxa7zxt2hbixylj6evi');
define('SECURE_AUTH_KEY',
'iotvmsflm6vetuj6fsaohmz7k0jag8l6kic549wte8bka0uknirx78416o3endpy');
```

```
define('LOGGED_IN_KEY',
'rdo6bhudamjh9oqu0kmv8i2zd0sqaaztoxfefebhgpoimwoe4sychwuw5agud260');

define('NONCE_KEY',
'pbdlf1fhvplgiwafb3z9d3meumxredptn3xsggcmvhknwn2pjpafgh8atstak32n');

define('AUTH_SALT',
'ejbxtgsir7n9k2juahn58qkhwqqnpzv0pvljxil3wkntkiemna2s1tm46pcybaof');

define('SECURE_AUTH_SALT',
'ej10q9upnayruulalkyd83bnz4dfmmhsubygxtqr4cgnb5yptowngk5mw21mttea');

define('LOGGED_IN_SALT',
'l9zpi2hxytfhczzuga3evrx06nc4p7l2z0ylb90gja18toed9qu4i9eptjlxrjay');

define('NONCE_SALT',
'f8fm9scvazlrbyt2axsqwf21tahvj9msibdljcpjle4z8rsnwakrxpeiho0lgmvo');


/**#@-*/


/**
 * WordPress Database Table prefix.
 *
 * You can have multiple installations in one database if you give each a unique
 * prefix. Only numbers, letters, and underscores please!
 */
$table_prefix  = 'wp_';


/**
 * WordPress Localized Language, defaults to English.
 *
 * Change this to localize WordPress.  A corresponding MO file for the chosen
 * language must be installed to wp-content/languages. For example, install
 * de_DE.mo to wp-content/languages and set WPLANG to 'de_DE' to enable German
 * language support.
 */
define ('WPLANG', '');


/**
 * For developers: WordPress debugging mode.
 *
 * Change this to true to enable the display of notices during development.
 * It is strongly recommended that plugin and theme developers use WP_DEBUG
 * in their development environments.
 */
define('WP_DEBUG', false);


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

```
/** Absolute path to the WordPress directory. */
if ( !defined('ABSPATH') )
      define('ABSPATH', dirname(__FILE__) . '/');

/** Sets up WordPress vars and included files. */
require_once(ABSPATH . 'wp-settings.php');
```

This is a goldmine for hackers. Just look at some of the things this file makes plain:

```
/** The name of the database for WordPress */
define('DB_NAME', 'samowen_wp344');

/** MySQL database username */
define('DB_USER', 'samowen_wp344');

/** MySQL database password */
define('DB_PASSWORD', '64rP6asS4p');

/** MySQL hostname */
define('DB_HOST', 'localhost');
```

This info gives the hacker all he needs to mount an SQL Injection attack and compromise your database. As the contents of posts and pages are stored in the database, this can seriously mess up your site!

But it gets worse...

if you look at the DB_NAME, the part before the _ (on this site: “samowen”) is usually the same log-in name people use for Cpanel, so your control panel is also vulnerable if a hacker can access this file. Also, many people use the same password for the database, WordPress admin., and Cpanel

Leaving this file unprotected is like buying an expensive car – a Mercedes Benz, say – driving it down to the worst neighbourhood in your town, parking it in a dingy side-street, rolling down all the windows, leaving the key in the ignition (or that electronic chip thing they have on expensive cars nowadays), going away, leaving it all night, and expecting the car to still be there in the morning. You might be lucky, but it's highly likely an opportunist car thief will decide to take it for a ride. The same applies here: An opportunist hacker won't be able to resist it.

If you go to most WordPress sites and type in the site URL followed by /wp-config.php at best you'll see a blank page, at worst you'll see all the info above. Once a hacker can access the file, it's frighteningly easy to to find out all this info. There is a way to hide both wp-config.php and wp-admin. I'll explain more later in this report. By the way: The file above is from a site I no

longer use. It will have been taken down before you read this, so I don't mind sharing it; however, I advise you to keep the information about your site confidential.

Plug-ins

Plug-ins are another way hackers can attack your site.

According to the US National Vulnerability Database, there are thousands of vulnerabilities in WordPress, and most of them are related to plug-ins. You can read the list here: https://nvd.nist.gov/products/cpe/search/results?keyword=wordpress&status=FINAL&orderBy=CPEURI&namingFormat=2.3

Some unscrupulous developers also build a “back door” into plug-ins so they can attack your site! It's a good idea to be wary of plug-ins from sources you don't know or trust. Only install plug-ins from trusted sources. Those you can install from wordpress.org are probably a safe bet, though. You should also keep plug-ins updated to the latest versions, too. Reputable developers will eliminate any vulnerabilities as soon as they are aware of them and issue an update accordingly.

Themes

The same applies to themes as applies to plug-ins. They are often a way-in to your site for hackers, and sometimes hackers will substitute a theme on your site with another one – complete with malicious code and bogus links.

Just like with plug-ins, some unscrupulous developers will also build a back-door into themes, so be wary of “free” themes from untrusted sources. You could end up paying a very high price indeed!

Server-side Vulnerabilities.

Finally, there are what I call “server-side vulnerabilities”. These are ways a hacker can attack your site by getting access to the server itself and attacking your site that way. This is a particular problem on shared hosting. You might take all the precautions in the world, but still find yourself hacked by a someone who has managed to exploit a vulnerability on another account or program. This is why you should notify your hosting company if you get hacked.

You should also check any other software you have running on your site for vulnerabilities – especially PHP based programs like help desks, forums, and IMAP and FTP servers. Keep them updated to the latest versions. If you find your site constantly being hacked despite the precautions I'm going to show you later on, your only recourse may be to take your business elsewhere and change hosting companies.

What to do if Your Site Gets Hacked

If you're like most people, the first thing you'll do is utter some unrepeatable expletive! After that, you'll need to do something more practical.

Log into Cpanel, or whatever control panel your hosting account comes with. If you can't access Cpanel (the hacker has changed the username and password, say) you'll need to get onto your hosting company as they will have to reset things from their side to let you in. You should notify your hosting company anyway as a matter of courtesy, as the hacker may be getting in through another site on the same server.

If you have your site backed up from within Cpanel, just use the backup/restore function to restore your site to its pre-hacked state. This is by far the easiest way to deal with a hacker attack. Then you can follow the procedures in the next chapter to keep the hackers from coming back. You'll lose any entries you've made since you last backed your site up, though.

If you don't have a backup, or you've made a lot of changes since the last backup, all is not lost. You can still try to get your site back. It's difficult and time consuming, but possible.

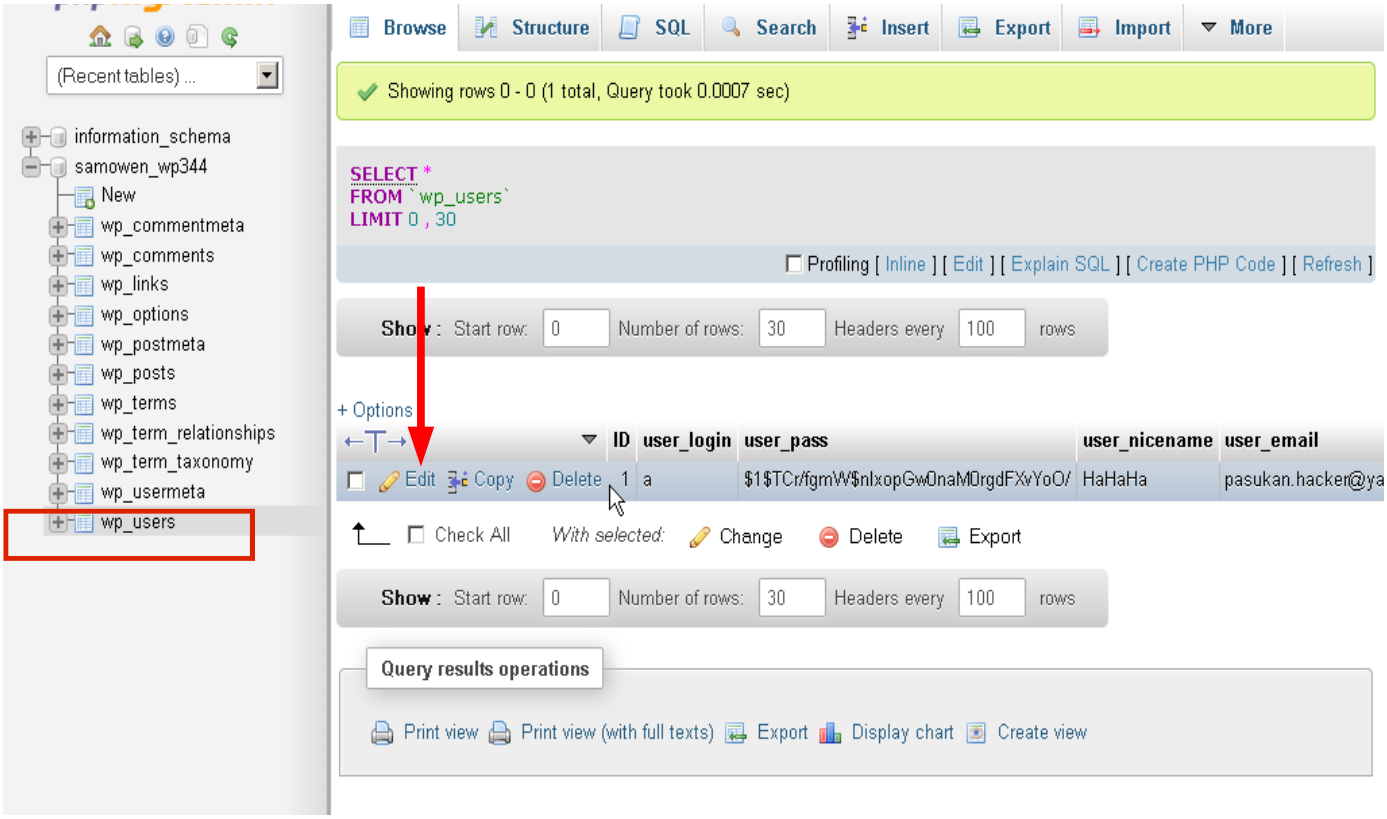
Here's what you can do...

If you're unable to access the WordPress Dashboard, it means the database has been compromised and the hacker has changed the username and password – that's what happened on all my sites.

About halfway down the Cpanel home page you'll find a section on databases. Look for the icon that says “phpMyAdmin”, and click on it. Choose the appropriate database from the list on the left. If you have more than one database, you'll have to check which database is the right one. You'll find it listed in wp-config.php.

On some other control panels, you have to click on the “Databases” icon, choose a database from the list, then click the button that says “Edit”. From that point on, though, the process is pretty much the same.

At the bottom of the stack, you'll find a database table called “wp_users”. Click on that. In the window to the right, you'll be able to see the information the table contains.



Chances are, you'll find someone else's e-mail address where it says “user_email” and the “user_login” and other information has been changed, too.

Click the “Edit” link and change user_login and other settings to ones you can remember (it's best not to use “admin” for the log-in). Change the e-mail address back to one you use, and click on “GO” or “OK” (depending on what version you have).

You can then go back to wp-admin and follow the instructions to change your password – it will send the e-mail to the right address now. You'll then be able to log-in to the WordPress dashboard and assess the damage.

Before you do that, though...

While you're still in Cpanel, you'll want to check on one or two other things. First of all, go through the files and folders (you can also use File Manager or third party FTP software like FileZilla). Look for rogue files and folders that

may have been placed on your server by the hacker. Pay particular attention to the date associated with the file or folder. If it is around the date of the hacking, regard it as suspicious. On my server, the hacker inserted a directory with a number of .php files. I deleted them all, but I still get error log readings showing 404 errors from a remote site where the hacker is probably still running a cron on another hijacked server trying to access them.

Which brings me to the next point...

Check through any Cron Jobs (AKA. Scheduled Tasks) that may be running. Make sure they're only running scripts on your site. Delete any tasks that look suspicious. Many times hackers will attempt to run scripts or bots on other sites/servers from your server. This uses up your server resources and can get your account suspended.

Once you're able to log back into the WordPress Dashboard, check to make sure your theme has not been hijacked. Goto Appearance > Themes. Activate another theme – either from the available themes on your site, or install another one from within WordPress by selecting the “Install Themes” tab.

Once the new theme has been activated, click on the “Visit Site” link and see if the site displays correctly. If it does, delete your old, hacked theme and reinstall it from a trusted source. If it doesn't work, then your core WordPress code may have been compromised, and you'll need to follow the steps below to reinstall it.

You'll also need to go through all your pages and posts to make sure they have not been interfered with. Use the HTML setting rather than the visual one to look for rogue lines of code and links to sites you've never heard of. This can take a long time if you have a lot of posts or pages on your site, but it is imperative you do a thorough job.

Next, go through all the pictures in the media library to see if they have been tampered with. Sometimes hackers hide malicious scripts in picture files, so if the file size seems too big, or the date has been changed, delete the file and reinstall from a trusted source.

Finally, you may want to delete all your plug-ins and reinstall them from trusted sources. If you have a shopping cart or a membership site this might not be practical (in which case you should check the code thoroughly) but otherwise it's a good idea as the .php code may have been interfered with.

If your core WP files have been compromised, you may have to re-install WordPress from scratch.

In which case, follow the steps below:

1. Backup your database from within Cpanel, and download it to your computer's hard drive.

2. Delete the WordPress installation. If possible do this by using the uninstall procedure from within Cpanel (if you have Fantastico or Softalicious this should be pretty straightforward). This way all the files, folders and database info will be completely removed from your server, and any “back doors” firmly closed.
3. Re-install WordPress. Use the same settings as before (I'll show you how to change them later) so you can be sure everything will work properly. Restore your database, plug-ins and theme. Your site should now be back to its pre-hacked state.

In a severe hacking attack some or all of these steps may not be possible, and you must prepare yourself for the worst case scenario: You may lose everything, and there's not a lot you can do about it.

Protect Yourself From Hackers

Disclaimer:

No system (not even the one I'm going to tell you about here) can 100% guarantee your site won't get hacked. If hackers are bound and determined to get into your site, they will. What you have to do is make life difficult for the opportunist hacker so they will decide your site is too much bother and leave you alone.

Your site may still be vulnerable to server-side hacking despite what I am going to show you here. If you find you're still getting hacked despite all the precautions I detail in this chapter, the problem may rest with your hosting company (particularly if you're on shared hosting) and your best recourse may be to take your business elsewhere.

You should keep vigilant – especially when it comes to installing themes and plug-ins from untrusted sources – and back up your site regularly.

After doing a lot of online research and going through a number of security plug-ins, I found a combination of two (free) plug-ins to protect my site. There are a lot of plug-ins out there which claim to protect your site from hackers; however, most of them are premium plug-ins which come at a hefty price or require an on-going subscription. The two that I have used address the vulnerabilities identified in Chapter 2.

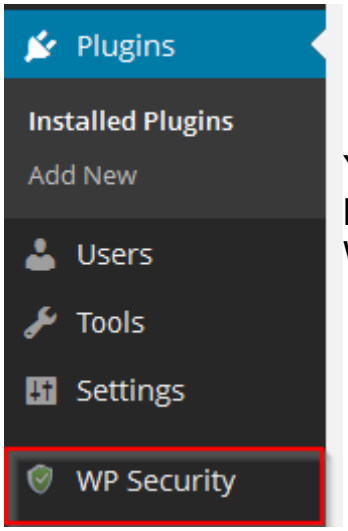
The plug-ins are: All In One WP Security & Firewall and Wordpress File Monitor.

Installing the plug-ins

Both these plug-ins can be installed from within the WordPress Dashboard.

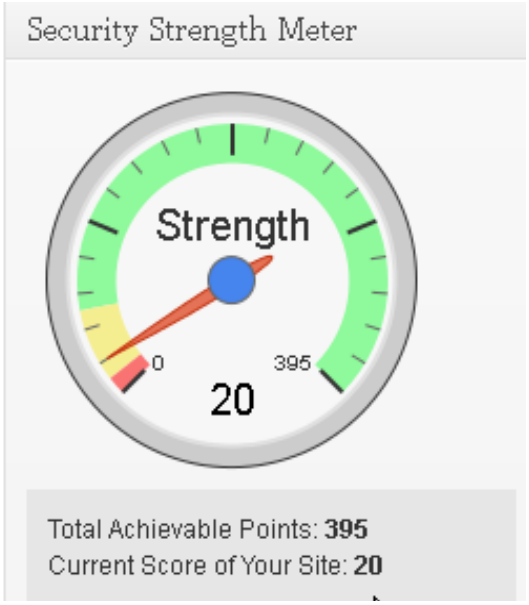
Goto “Plugins” and “Add New”. In the search box type in “All In One WP Security & Firewall” or “Wordpress File Monitor” (without the quotes). I recommend you install the plug-ins on at a time. In this chapter, I'm going to install the All In One WP Security & Firewall plug-in first, and the WordPress File Monitor later.

When the search has found the correct plug-in, Click “Install now”, follow the on-screen instructions, and click “Activate”.

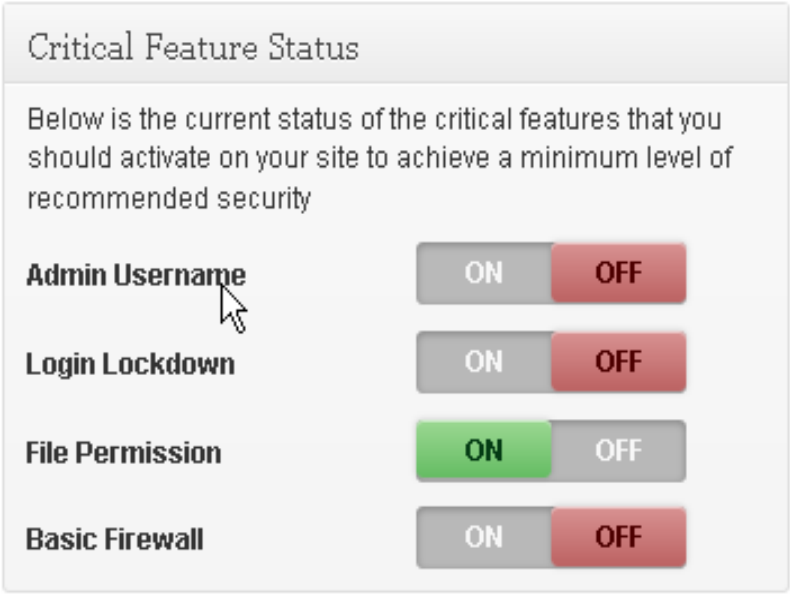


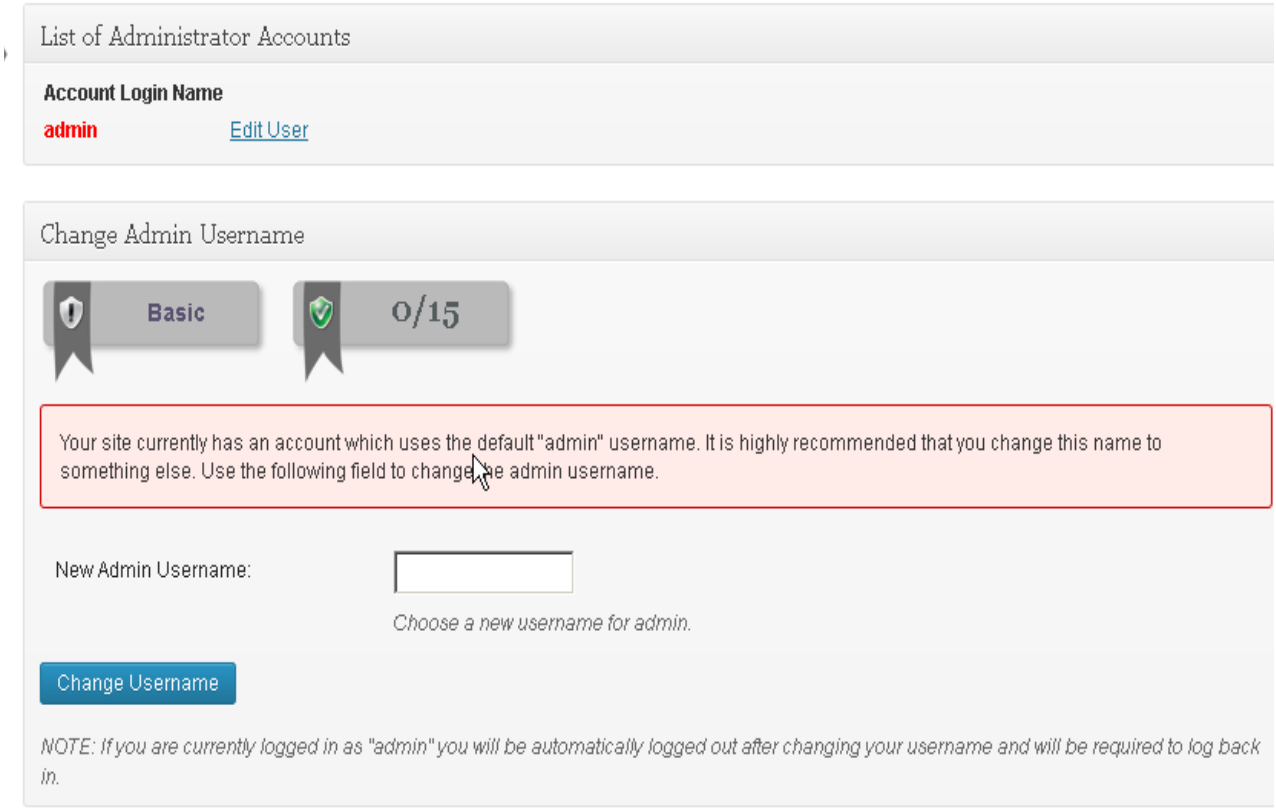
You'll find a new tab titled “WP Security” at the lower left-hand side of your screen. Click on this tab to open the WP Security Dashboard.

The dashboard will display a security strength meter. As you can see from the picture below, at just the default settings, the security isn't very strong! We'll come back to the meter from time to time throughout this chapter as it's a good way of monitoring how your security is progressing.



Below the meter, you'll see “switches” for the Critical Features Status. These are settings you should activate to give yourself the minimum level of protection from hackers.





If the username is “admin” (the default), type the new admin username in the box, then click the “Change Username” button. This just changes the username, NOT the display name – so your posts will still display as “admin”

completely foxing any potential hacker. Clever!

If you're logged in as “admin”, you'll automatically be logged out after you make the change, and you'll have to log in again under your new username.

15 Points added to your security score!

Go back to the dashboard and activate Login Lockdown.

Login Lockdown

This is your first line of defence against brute force attacks. If anyone unsuccessfully tries to log-in to your site a maximum number of times within a certain period of time, they are locked out.

The default settings allow three unsuccessful login attempts within 5 minutes from the same IP address. After that, the potential hacker is locked out of your site for an hour.

You set things up so that you get sent an e-mail to notify you of a lockout. This e-mail will tell you what username was trying to log in, the date and time of the potential brute force attack, and the IP address of the potential hacker. You can then blacklist the IP address and block them out permanently.

If you allow others to log-in to your website, I recommend you check the checkbox to display a generic error message to tell people they've been locked out, why they've been locked out, and when they'll be allowed to log-in again. That way, you're not going to upset genuine users.

There is also a checkbox you can check to instantly lockout anyone who tries to log in with an invalid username. Again, if it's just you who logs into the site, and you allow your web browser to store the username and password, it's worthwhile checking this. Then, if someone tries to log in with the username “admin” (say) they are automatically locked out, and you'll be notified. If you allow others to log in – to comment on posts for example – it's best to leave this unchecked as you don't want to inadvertently lock out genuine users who might enter a typo.

You might want to allow more than 3 login attempts, or make the time frames longer or shorter, it's up to you. For me the default settings are just fine. Make your choices and click the “save settings” button.

Another 20 points added to your security score!

Before you leave this page and go back to the dashboard, scroll up to the top of the page, and click each of the tabs in turn.

Login Captcha does exactly what it says: It adds a captcha (a sum, actually)

to the log-in and lost password pages, so whoever comes to the page has to prove they're human. This discourages bots. Check the check boxes, click the “Save Settings” button, and move on to the next tab: Login Whitelist.

You may or may not want to activate this setting. Basically, this gives you the option of only allowing certain IP addresses or ranges to have access to your log-in page. Any IP addresses not listed in the listing box within this tab will be blocked out.

Use this setting with caution. If you access your WordPress site from more than one computer, or if (like me) your ISP routes your access via several servers, so you sometimes log in from a different IP address, you might want to give this setting a miss. If you only ever log-in from one computer on a dedicated IP address, then by all means check it as it gives you an extra layer of protection.

The setting modifies your .htaccess file, so be sure to back the file up before making any changes. That way, you can restore it and gain access if things go wrong.

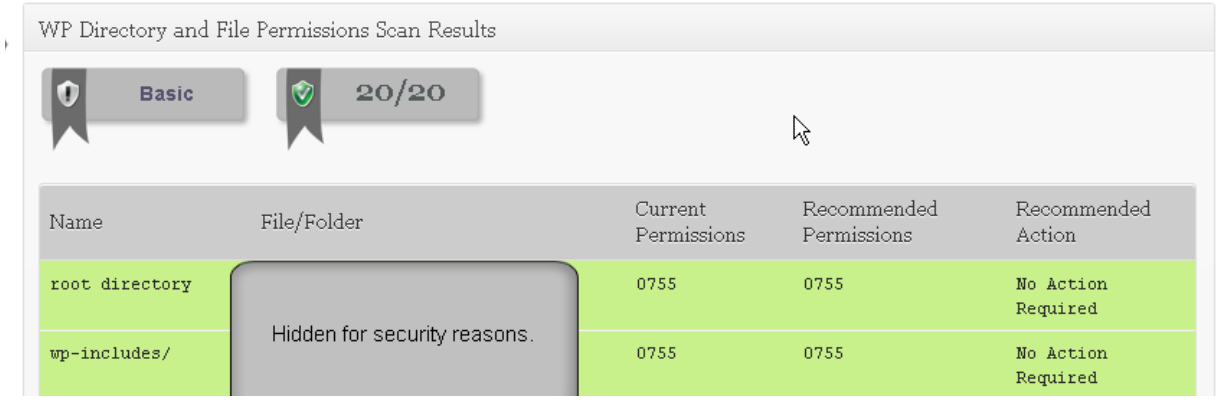
Failed Login Records is more of a tool than a setting. It displays a list of failed log-in attempts for your site. This can come in handy if you need to carry out security investigations because it will show you the username, IP range,ID (if applicable), and date of every failed log-in attempt.

Force Logout sets an expiry time for every admin session. After the specified number of minutes, the session will time out, and the administrator will have to log-in again. This is useful if you access the admin area from a computer others may have access to. If you leave the computer unattended, you'll be logged out automatically. It's adds 5 security points to your overall score.

Account Activity Logs is another useful tool. It displays the login activity for registered admin accounts on your site. It shows the last 50 logins, their username, IP Address, and the time they logged in and out. If someone has stolen your username and password, it will show up here.

Logged In Users displays a list of everyone currently logged in to your site. If you suspect someone is using your site who shouldn't be there, you can get their IP address and block them out by adding it to your blacklist.

With all the Login Lockdown features set, go back to the dashboard and select the next “switch”: File Permission.

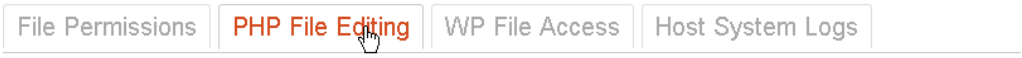


The WordPress file and folder permission settings govern the accessibility and read/write privileges of the files and folders which make up your WP installation. On my server, the basic permissions were already set correctly (hence the score of 20 when the plug-in was first installed) and WordPress comes with reasonably secure file permission settings for the file system it installs.

However, sometimes people or other plug-ins modify the various permission settings of certain core WP folders or files, and in so doing, end up making the site less secure because they chose the wrong permission values.

This feature scans the crucial WP core folders and files, and highlights any insecure permission settings. Keep a particular lookout for files or folders with permissions set to 0777 as this can be a back door for hackers, and could indicate a file or script being executed from another site.

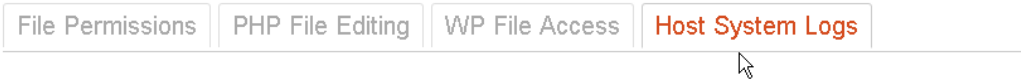
Scroll up to the top of the page, and click on the tabs to add extra layers of file security.



Check the checkbox and click the “Save Settings” button. Another 10 points will be added to your overall security score.

The next tab is WP File Access. This feature allows you to prevent access to files such as readme.html, license.txt and wp-config-sample.php which are delivered with all WP installations. By preventing access to these files you are hiding some key pieces of information (such as WordPress version info) from potential hackers.

Check the checkbox and click the “Save Settings” button. Again, another 10 points will be added to your overall security score.



Depending on the nature and cause of the error or warning, your hosting server can create multiple instances of this file in numerous directory locations of your WordPress installation. By occasionally viewing the contents of these logs files you can keep informed of any underlying problems on your system which you might need to address.

Keep a particular lookout for any PHP files that seem to be executed independently of any other files. This can indicate a hacker running a remote script on your server.

With all the file permissions taken care of, go back to the dashboard and select the final “switch” Basic Firewall.

Basic Firewall

The first tab you'll come to contains two settings, each of which adds 15 points to your overall security score. I recommend you activate both of them (although there is a caveat for the second).

The first one will enable Basic Firewall Protection.

This setting will implement the following basic firewall protection mechanisms on your site:

1. Protect your .htaccess file by denying access to it.
2. Disable the server signature.
3. Limit file upload size to 10MB.
4. Perhaps most importantly: Protect your wp-config.php file by denying access to it.

The features are applied via your .htaccess file and should not affect your site's overall functionality. However, if you have not already done so, you should backup your active .htaccess file just in case.

The second enables WordPress Pingback Vulnerability Protection.

You should use it if you are not using the WP XML-RPC functionality, and want to enable protection against WordPress pingback vulnerabilities

This setting will add a directive in your .htaccess to disable access to the WordPress xmlrpc.php file which is responsible for the XML-RPC functionality such as pingbacks in WordPress.

Hackers can exploit the various pingback vulnerabilities in the WordPress XML-RPC API in a number of ways such as:

1. Denial of Service (DoS) attacks
2. Hacking internal routers.
3. Scanning ports in internal networks to get info from various hosts.

Apart from the security protection benefits, this feature may also help reduce the load on your server, particularly if your site currently has a lot of unwanted traffic hitting the XML-RPC API on your installation.

Check the checkboxes and click the “Save Basic Firewall Settings” button.

Just like on the other settings, there are some tabs at the top of this page. Click each of them in turn to adjust the settings.

The first one, Additional Firewall Rules has 5 settings. They enable you to activate some of the more advanced firewall settings to your site.

The advanced firewall rules are applied via the insertion of special code to your currently active .htaccess file.

Due to the nature of the code being inserted to the .htaccess file, this feature may break some functionality for certain plug-ins and you are advised to backup your .htaccess before applying this configuration.

The first setting is Listing of Directory Contents. By default, an Apache server will allow the listing of the contents of a directory if it doesn't contain an index.php or index.html file. Activating this setting will prevent the listing of contents for all directories. On most sites you'll want to check this; however, if you're running a membership site, have a forum or shopping cart on your site – any site where you want as many pages listed as possible – it's best to leave this unchecked.

NOTE: In order for this feature to work "AllowOverride" must be enabled in your httpd.conf file. Ask your hosting provider to check this if you don't have access to httpd.conf

The second setting is Trace and Track, and if selected will disable the feature. HTTP Trace attack (XST) can be used to return header requests and grab cookies and other information. This hacking technique is usually used together with cross site scripting attacks (XSS). Disabling trace and track on your site will help prevent HTTP Trace attacks.

The third setting will forbid Proxy Comment Posting. This setting will deny any requests that use a proxy server when posting comments. By forbidding proxy comments, you are in effect eliminating some SPAM and other proxy requests.

The fourth setting will Deny Bad Query Strings. This feature will write rules in your .htaccess file to prevent malicious string attacks on your site using XSS. Use this setting with caution as some of these strings might be used for plug-ins or themes and enabling it might break some functionality.

The final setting enables Advanced Character String Filtering. This helps prevent malicious string attacks on your site coming from Cross Site Scripting (XSS). This setting matches common malicious string patterns and exploits and will produce a 403 error for the hacker attempting the query.
Again, use with caution as some strings for this setting might break some functionality.

Decide which settings you want to activate, and click the “Save Additional Firewall Settings” button. Then move back to the top and select the 6G Blacklist Firewall Rules tab.

This feature allows you to activate the 6G firewall security protection rules designed and produced by Perishable Press.

The 6G Blacklist is a simple, flexible blacklist that helps reduce the number of malicious URL requests that hit your website.

The added advantage of applying the 6G firewall to your site is that it has been tested and confirmed by the people at PerishablePress.com to be an optimal and least disruptive set of .htaccess security rules for general WP sites running on an Apache server or similar.

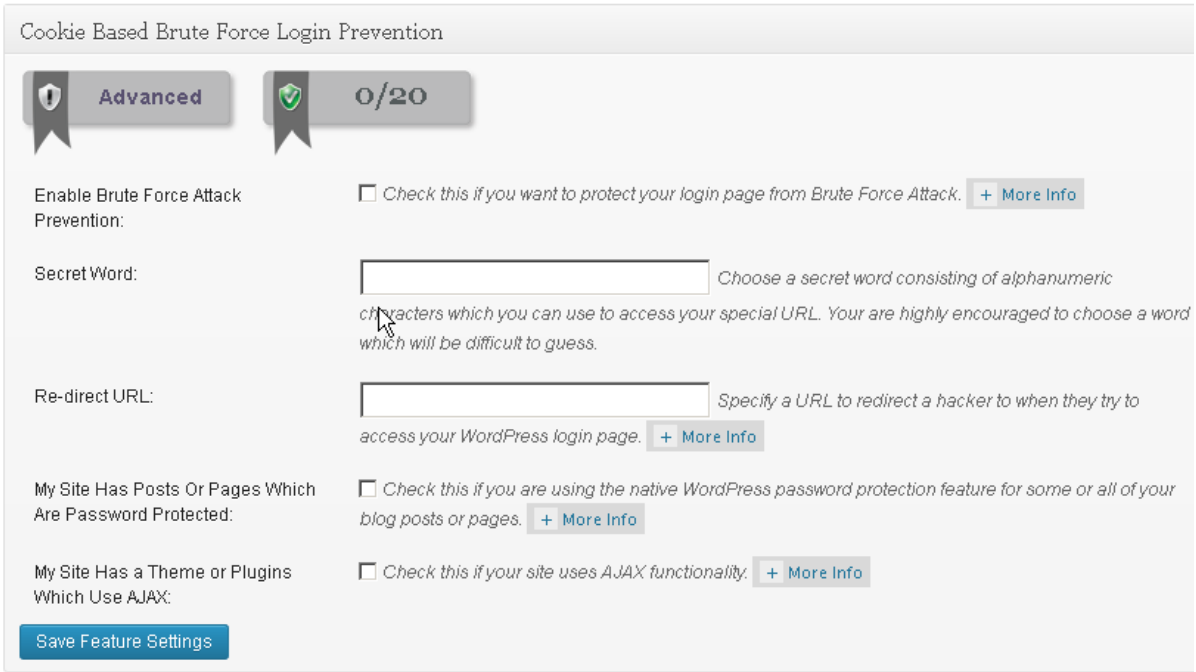
Amongst other things this setting will block forbidden characters commonly used in exploitative attacks, block malicious encoded URL characters such as the ".css(" string, guard against the common patterns and specific exploits in the root portion of targeted URLs, and will disallow illicit characters to stop attackers from manipulating query strings. You can find out more by visiting the Perishable Press website http://perishablepress.com/ . It's well worth activating this feature, and it adds another 20 points to you overall score.

Check the checkbox and click the “Save 6G Firewall Settings” button, then go back to the top of the page and select the next tab: Brute Force prevention.

I strongly recommend you implement this setting.

Basically, what this setting does is create a special log-in URL for the wp-admin area. This special log-in incorporates a secret word. Anyone who doesn't know this secret log-in cannot access the wp-admin log-in page and is redirected somewhere else. This gives you as close to total protection from brute-force log-in attempts as it's possible to get.

Take care when choosing your secret word (a combination of words and numbers is probably best). Don't choose anything related to your site topic or anything that's going to be easy to guess.



If your site has pages or posts which are password protected, or uses AJAX themes or plug-ins, you'll want to check the appropriate checkboxes, and then click the “Save Feature Settings” button.

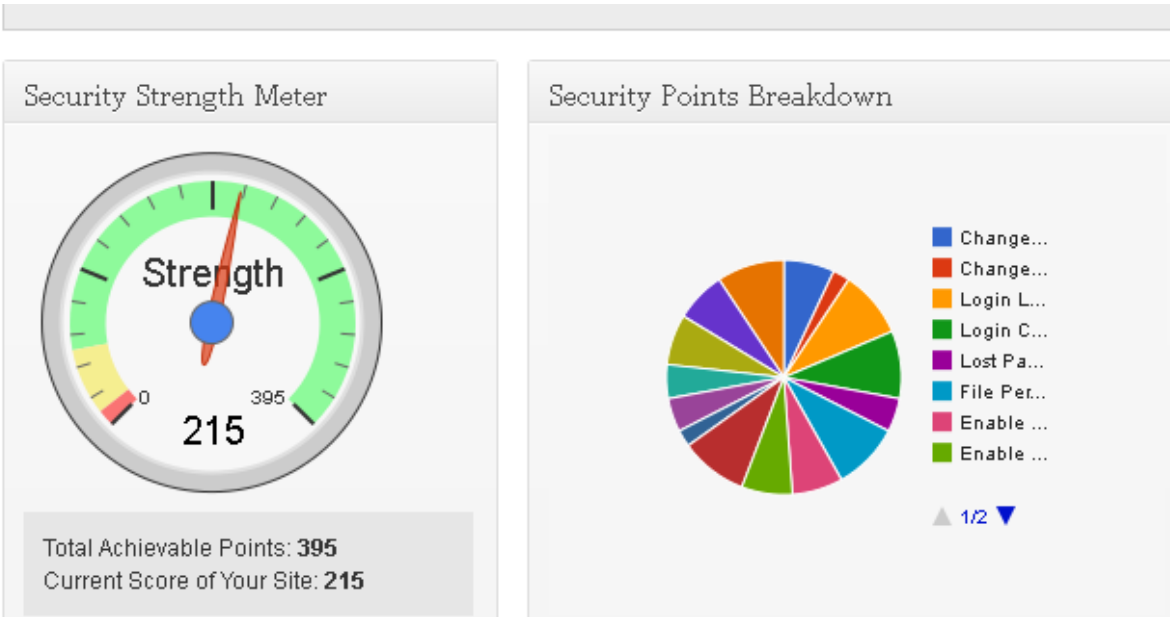
The plug-in will generate a log-in URL which looks something like this:

http://yoursite.com/wpsecurity?jenny8675309-1

You will need to log-in using that URL from now on. Write it down and keep it in a safe place. I recommend you bookmark the URL in your web browser and/or create a desktop shortcut which you can back-up onto a backup drive or USB memory stick.

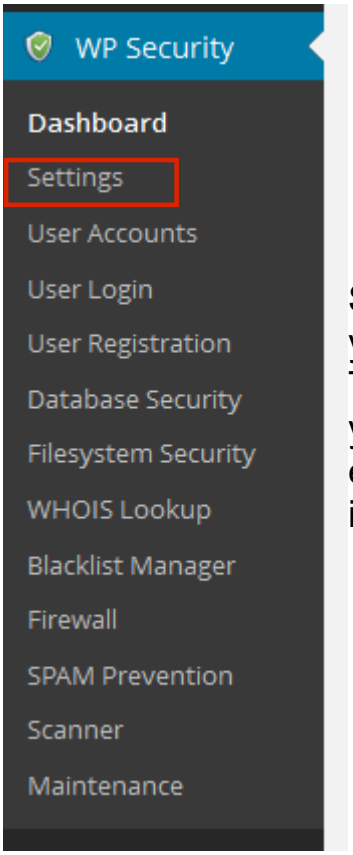
If you forget the URL you will not be able to log-back into your WordPress Dashboard without restoring your .htaccess to its previous state.

With all that done, go back to the security dashboard.



You'll notice that just by applying the main settings, your security score has gone up dramatically! You'll also notice a pie chart breakdown showing you how each step has contributed to your score.

There is still room for improvement, though. Let's look at some of the other features of this plug-in.



Second from the top is the *Settings* area. From here you can disable the firewall, security features, or both. This comes in handy is you are having problems with your site's functions after getting everything set up. This enables you to pinpoint where the trouble is, and makes it easier to correct.

From the tabs at the top of the page, you can access your site's .htaccess file, wp-config.php file, and back up and restore them without having to log into Cpanel.

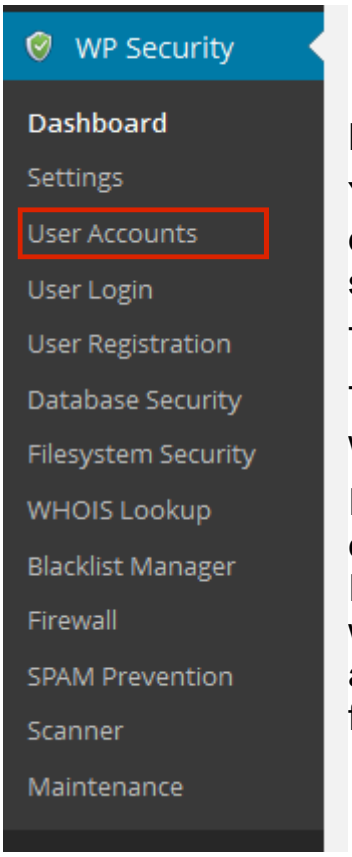
The tab at the far right enables you to hide some of the meta tag info generated by WordPress.

The WordPress generator automatically adds some meta information inside the "head" tags of every page on your site's front end. For example:

```
<meta name="generator" content="WordPress 4.7.5" />
```

This shows which version of WordPress your site is currently running and can help hackers or crawlers scan your site to see if you have an older version of WordPress or one with a known exploit. Activating this feature will allow you to remove the WP generator meta info from your site's pages.

Check the checkbox, and click on “Save Settings”.



Next on this list is *User Accounts*.

You will have already set the settings in the first two tabs during the initial set-up from the dashboard, and shouldn't need any adjustment.

The third tab will test the strength of your password.

Type your password into the box and see how long it would take a hacker to crack it.

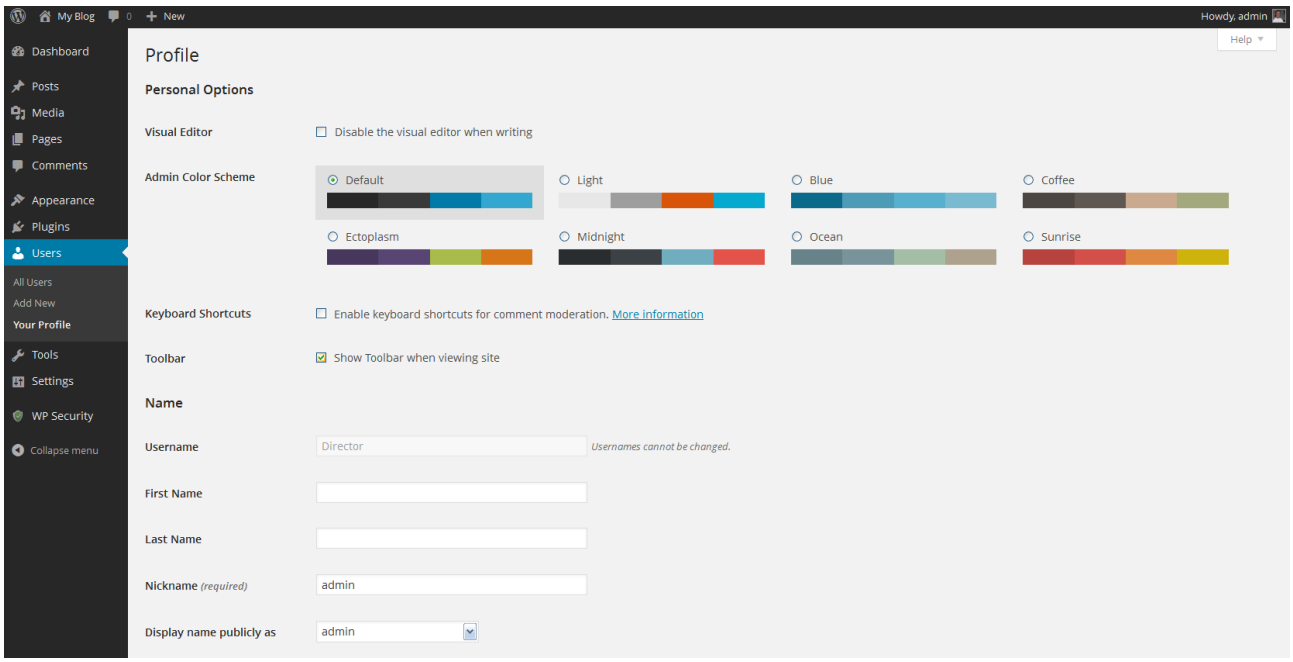
It's best to choose a really secure password – a combination of letters numbers and symbols works best. If you're in the UK or Eurozone, and have a keyboard with a Pound (£) or Euro (€) symbol key on it, then try and incorporate those into your password as a hacker from elsewhere is unlikely to have easy access to them.

If you need help coming up with a secure password, there are a number of websites online to help you. Just do a search in Google for “free password generator” and you'll find plenty. One I like is http://freepasswordgenerator.com/ . It's very easy to use and can generate a password for you very quickly.

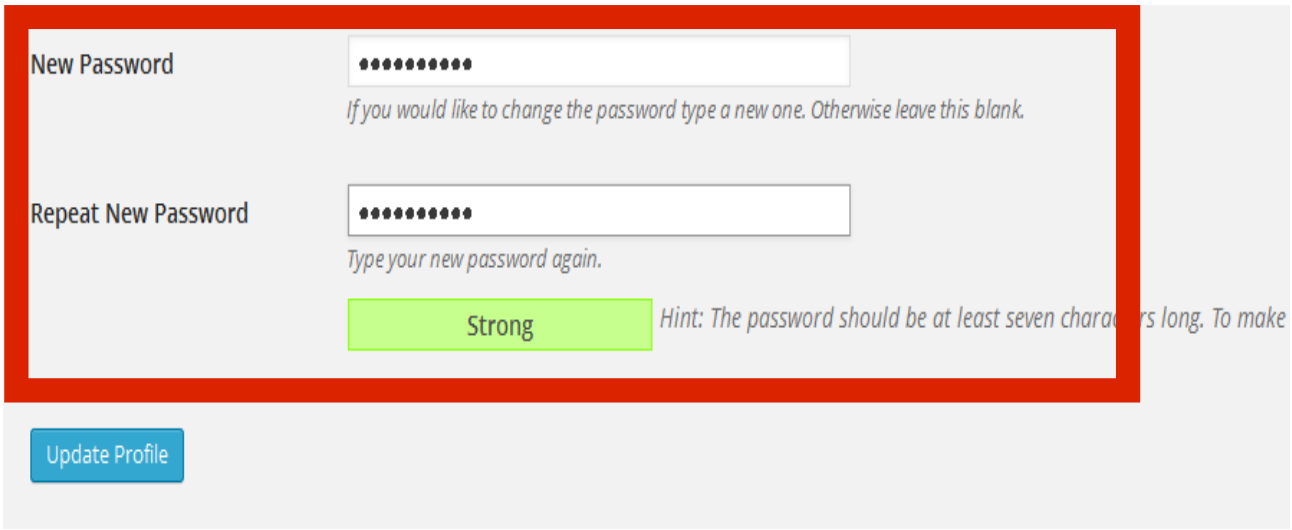
Copy and paste the new password into the strength meter to see how strong it is. The one I chose for my site would take a desktop PC approximately 9 years 6 months to crack. Now that's more like it!

Be sure to write you new password down, or paste it into a text file and keep it somewhere safe – otherwise you'll get locked out of your site and will have to reset the password all over again.

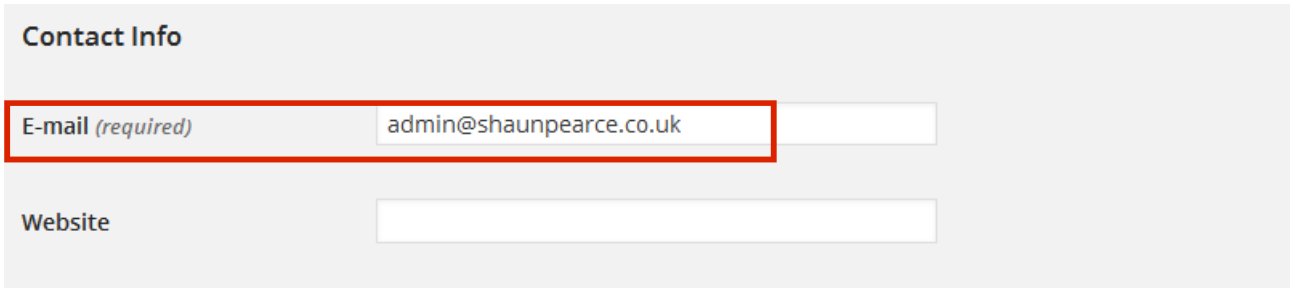
You'll need to change the password manually by editing your profile. You can access this one of two ways: Either by clicking in the top right-hand corner where it says “Howdy, [display name]” or by selecting Users > Your Profile from the menu on the left. Either way, you'll be taken to a page like the one below:



Scroll down to the bottom, and you'll see a place to enter your password.



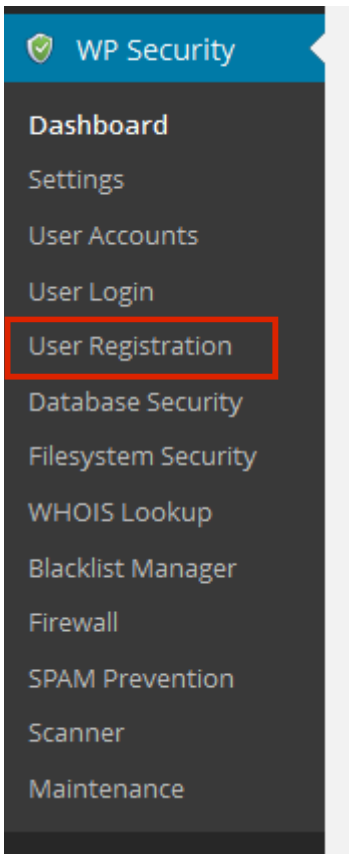
Copy and paste your new password. Before you click the “Update Profile” button, there's one more piece of housekeeping...



By default, WordPress sets the Administrator's e-mail address as “admin@[your URL]. This is very attractive to spammers, who may target your site, or attempt to manipulate the password reset function. For extra

security, change it to something else – preferably one that has no reference to your site – so don't choose “yoursitename@gmail.com” or similar. If you use a gmail or hotmail address, choose one people will find hard to guess.

Now click the “Update Profile” button, and head back to the security settings area.



User Login has already been dealt with during the initial setup, so the next area to turn your attention to is *User Registration.*

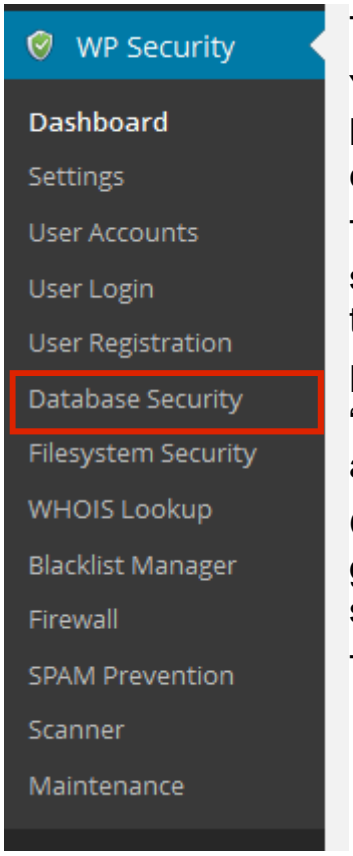
The first tab here is manual registration, and if you allow people to create their own accounts on your site via the WordPress registration form, you can minimize spam or bogus registrations by manually approving each registration.

This feature will automatically set a newly registered account to "pending" until you activate it. This way, anyone who has registered for the purposes of making spammy comments will be unable to log in without your express approval.

You can view all accounts which have been newly registered via the handy table on this page, and you can also perform bulk activation/deactivation/deletion tasks on each account.

The next tab enables you to put a Capcha (actually a mathematical equation) on to the registration page. This means whoever tries to register on your site will have to prove they're human, and helps to discourage spam bots.

Each of these settings will add another 20 points to your overall security score.



The next setting is *Database Security.*

Your database is your website's most important asset because it contains vital information. If it gets compromised, your website cannot function properly.

The database is also a target for hackers via methods such as SQL Injections and malicious code which targets certain tables.

By default, WordPress sets the database table prefix to "wp_". Hackers know this, and use this vulnerability to attack your site.

One way to reduce the chances of your database getting hijacked is to change the table prefix to something else.

Two notes of caution here:

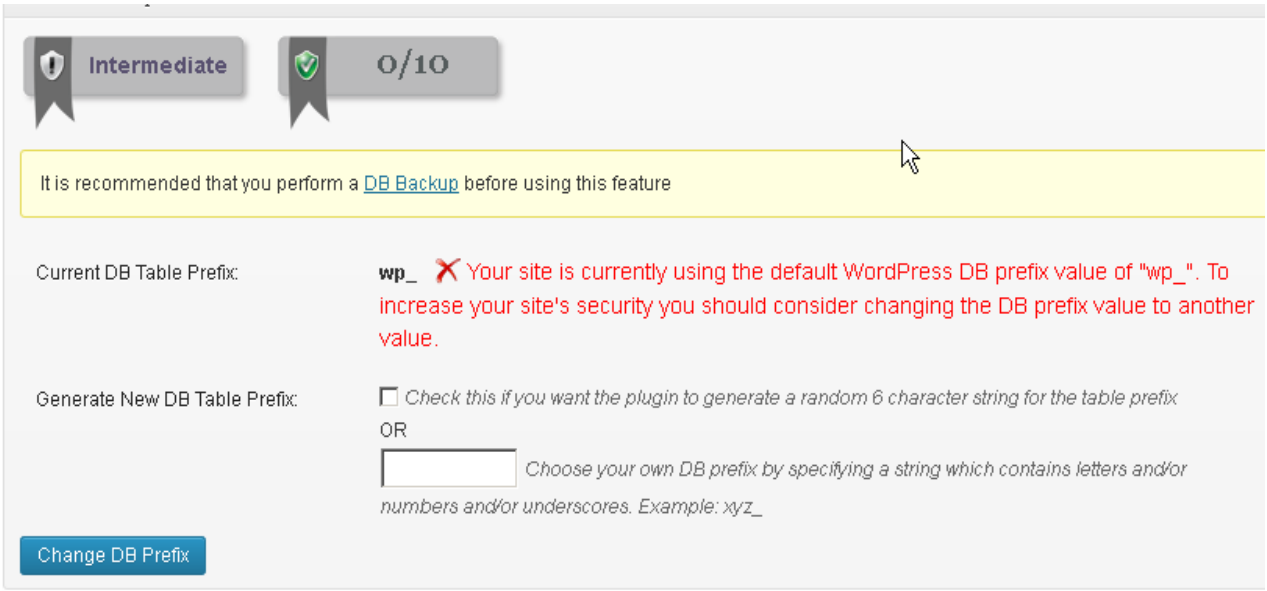
1. Backup the database first! You can do this by selecting the "DB Backup" tab at the top of the page and clicking on the "Create DB Backup Now" button. You can also schedule a DB backup at regular intervals – highly recommended!
2. If you are already running a plug-in which creates tables in your database, (a membership site, forum or shopping cart, say) it might be best to skip this as the plug-in may no longer work. If it's a new installation, I don't think you'll have any problems, but be sure to back-up just in case. One of my sites was already running s2 Member, and when I changed the prefix, the site crashed.

With your database backed up, go back to the first tab. There are two ways you can set a new database prefix:

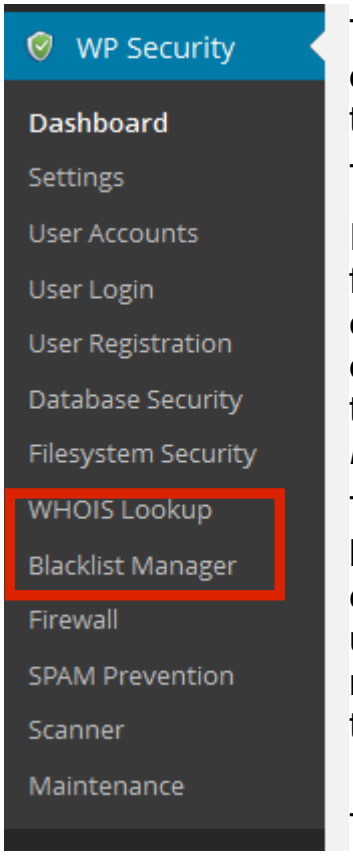
1. Select one yourself, or
2. Let the plug-in create one for you.

Personally, I recommend Method 2 as it's likely to be random and therefore more secure.

Click the "Change DB Prefix" button to implement this change.



Another 10 points will be added to your overall security score.



The next setting, *Filesystem Security*, has already been dealt with during the initial set-up, so let's move on to the next two: *WHOIS Lookup* and *Blacklist Manager.*

These are more tools than settings and work together.

If you find you're getting a lot of spammy comments, failed log-in attempts, or other suspicious activity, you can look up the domain or IP address in the WHOIS database from within WordPress itself. Once you have the IP address, you can blacklist them in *Blacklist Manager.*

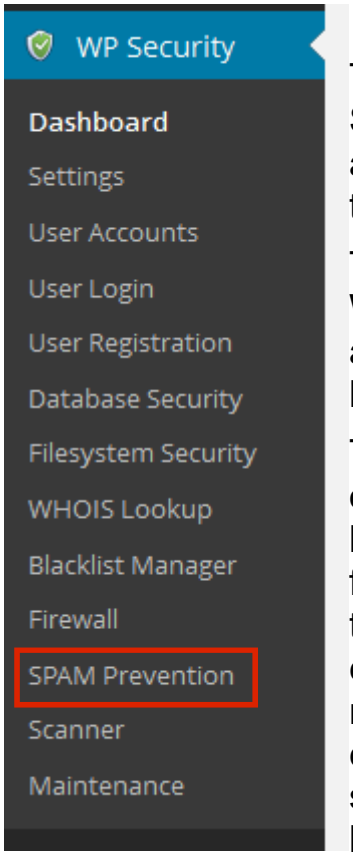
This feature gives you the option of banning certain host IP addresses or ranges, and also lets you ban certain user agents. It will deny total site access for users which have IP addresses or user agents matching those you have configured in the settings on the Ban Users page.

The plug-in achieves this by making appropriate modifications to your .htaccess file.

By blocking people via the .htaccess file, you are using the most secure first line of defence as it denies all access to blacklisted visitors as soon as they hit your hosting server.

Be advised, though. Many hackers use proxy servers or other means of masking their real IP address to counteract precautions like these.

The *Firewall* settings have been covered in the initial setup, so let's move on to the next setting: *Spam Prevention*.



The first tab you'll come to deals with the Comment Spam settings. There are two checkboxes, and if you allow comments on your blog, I advise you to check them both.

The first one, adds a simple CAPCHA field into the WordPress comments form. Anyone coming to your site and making a comment will have to prove they are human.

The second box blocks out spam bots. A large portion of WordPress blog comment spam is mainly produced by automated bots and not necessarily by humans. This feature greatly minimizes the useless and unnecessary traffic and load on your server resulting from spam comments by blocking all comment requests which do not originate from your domain. In other words, if the comment was not submitted by a human who physically submitted the comment on your site, the request will be blocked.

If you're manually approving comments too, this should mean the end of the line for comments spam.

The second tab, *Comment SPAM IP Monitoring,* displays a list of IP addresses of people or bots who have left spam comments on your site. This information can be handy for identifying the most persistent IP addresses or ranges used by spammers.

By inspecting the IP address data coming from spammers, you will be in a better position to determine which addresses (or address ranges) you should add to your blacklist and block.

To add one or more of the IP addresses displayed in the table on this page to your blacklist, simply click the "Block" link for the individual row, or select more than one address using the checkboxes, and then choose the "block" option from the Bulk Actions drop-down list and click the "Apply" button.

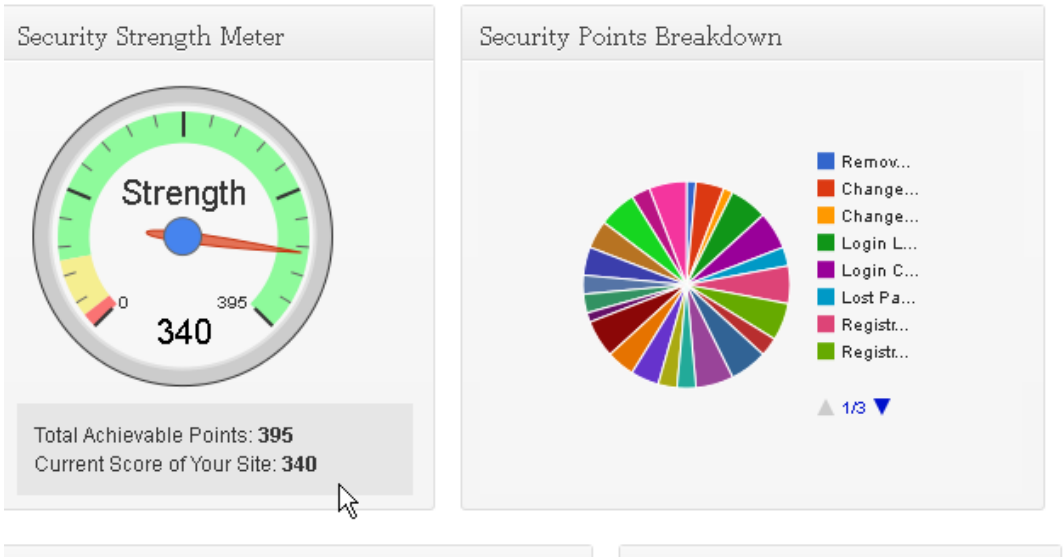
All these settings add another 30 points to your overall security total.

The next setting, *Scanner*, is the only part of this plug-in I didn't like very much.

This plug-in does the job well enough; however, I like the way WordPress File Monitor does it better – but that's just my personal preference. I'll cover the settings in WordPress File Monitor later in this chapter. For now, let's move onto the next setting: *Maintenance.*

This feature allows you to put your site into "maintenance mode" by locking down the front-end to all visitors except logged in users with super admin privileges.

Locking your site down to general visitors can be useful if you are investigating some issues on your site or perhaps you might be doing some maintenance and wish to keep out all traffic for security reasons.



To get better scanning – which may warn of a possible server-side hacking attack – I'm going to install the WordPress File Monitor plug-in.

WordPress File Monitor

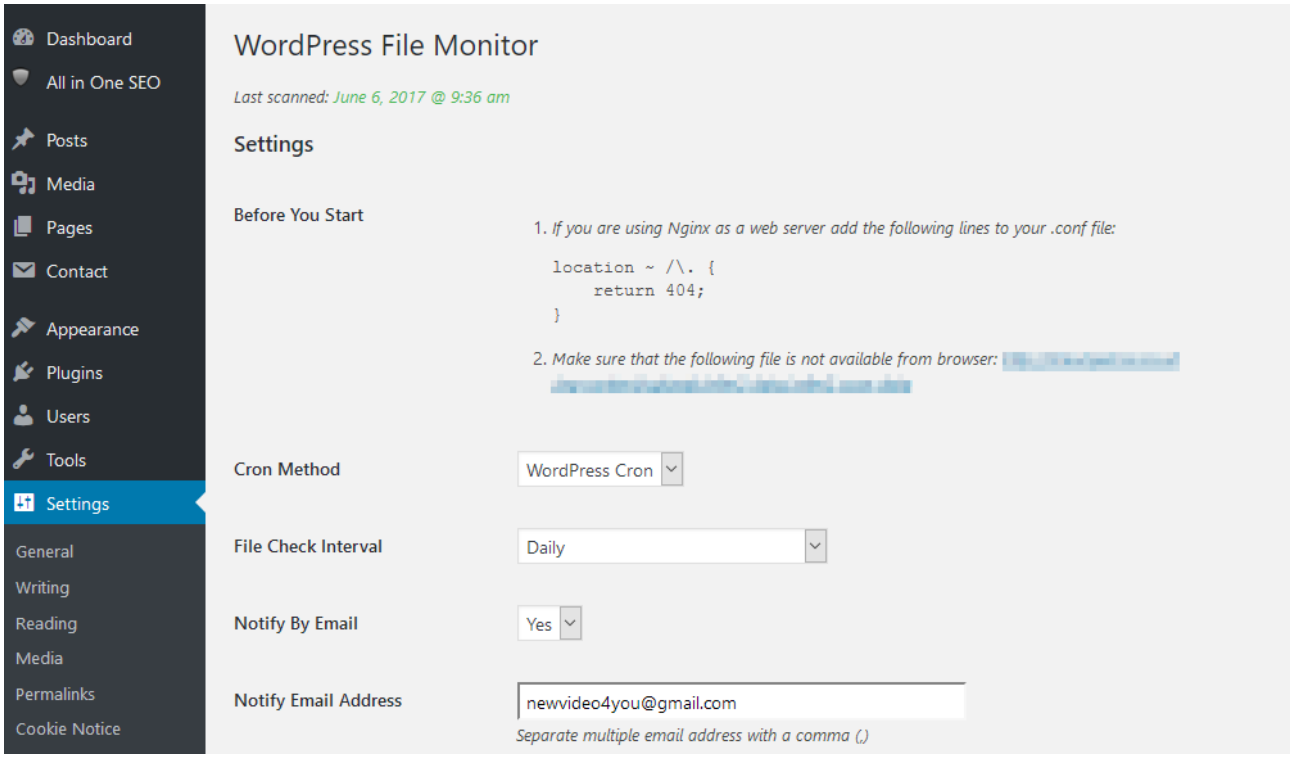
Given the opportunity, hackers can insert code or files into your system and carry out malicious acts on your site. Being informed of any changes to your files can be a good way to quickly prevent a hacker from causing damage to your website.

In general, WordPress core and plug-in files and file types such as ".php" or ".js" should not change often, and when they do, it is important that you are made aware when a change occurs and which file was affected.

The plug-in works by comparing the current state of your site to the previous scan and noting any differences. You may get some “false positives” when you update a plug-in, theme, or WordPress version, but it's a small price to pay for peace of mind.

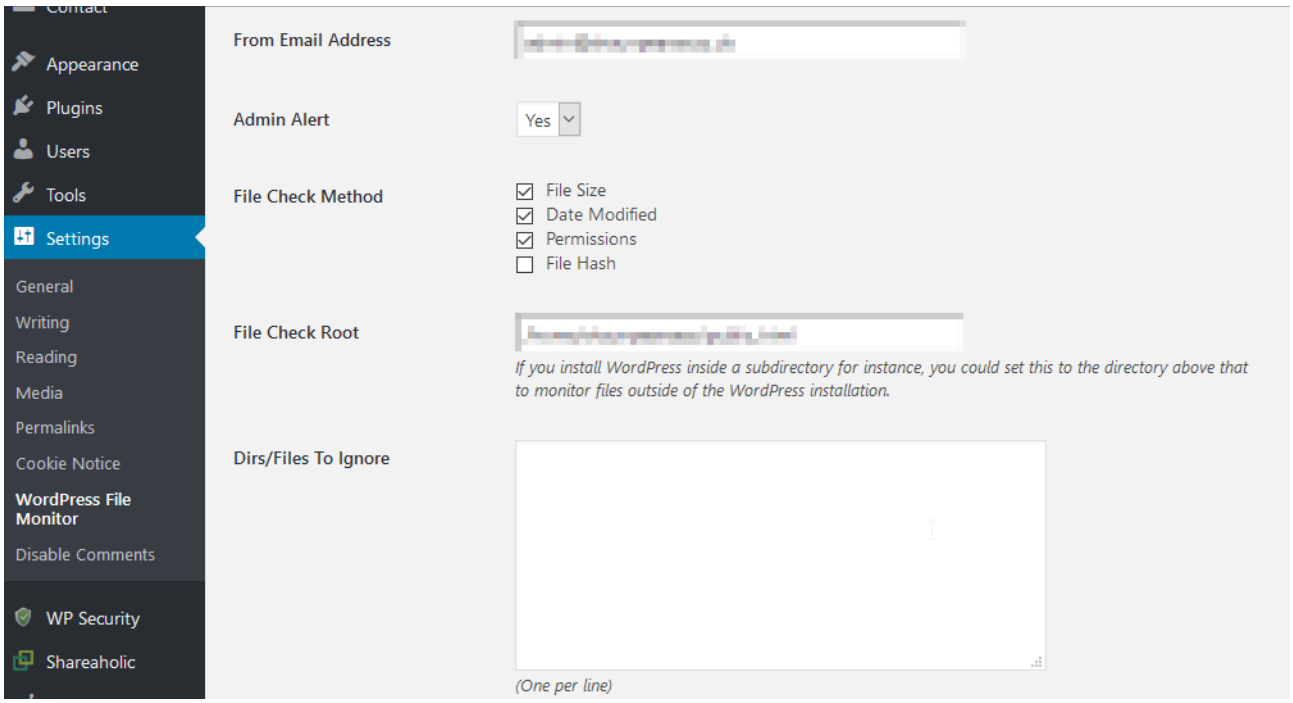
The plug-in installs from the WordPress Dashboard, and can be accessed via the *Settings* menu.

The setup is pretty straightforward:

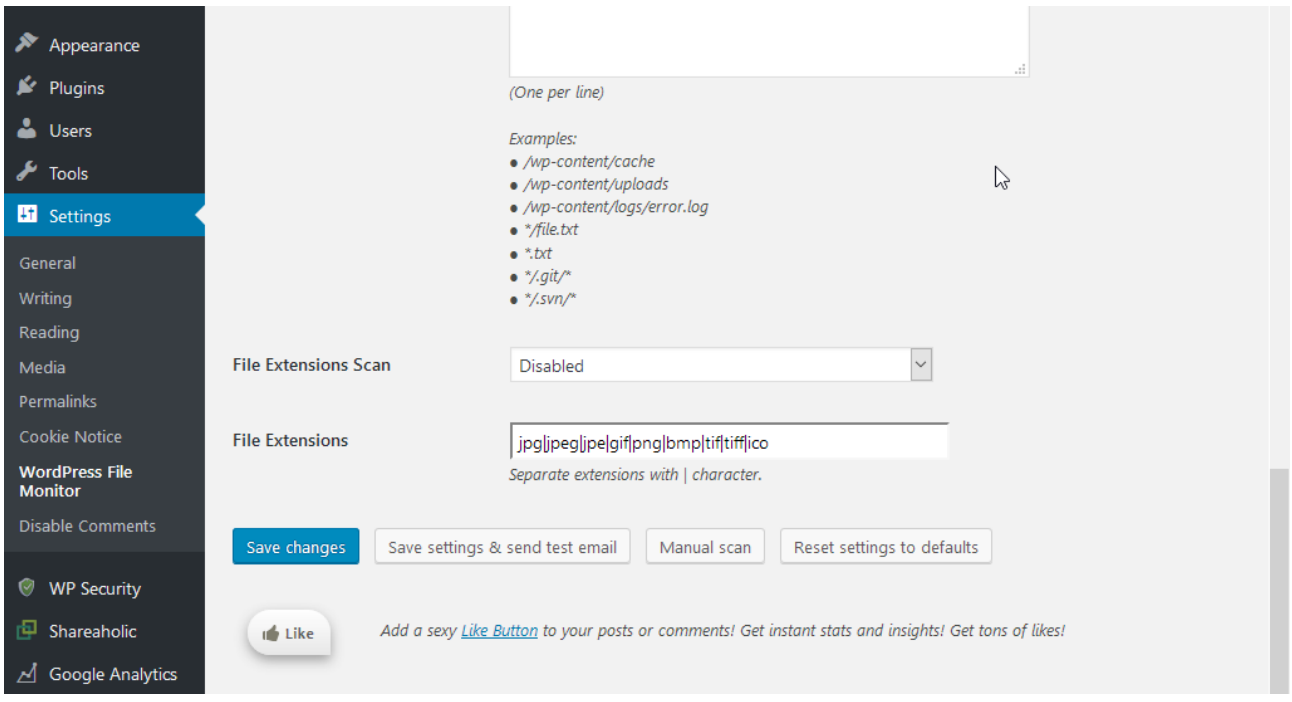


The plug-in operates via a Cron (which you can run from within WordPress or via Cpanel).

You set the file check interval from the drop-down menu. This can be up to twice daily if you wish, although for most sites, daily or weekly is fine. You set the “Notify” e-mail address, and the “From” e-mail address. Set the Notify By Email to “Yes” (that's the whole point after all).



You can check files by file size, date modified, permissions or file hash. I recommend you check at least the first three checkboxes. You can modify the file check root (which I'm hiding for security reasons as this has been installed on a real, live site).



You can specify certain files or directories to ignore, and you can also specify that only files with certain extensions be scanned. This can cut down your server load if you have a large site.

Once everything has been set up, click the “Save Changes” button, or the “Save settings & send test email” button.

You'll then need to run a manual scan (by clicking the “Manual Scan” button) so the plug-in has something to reference against, and you're done.

A couple of other housekeeping tasks to do...

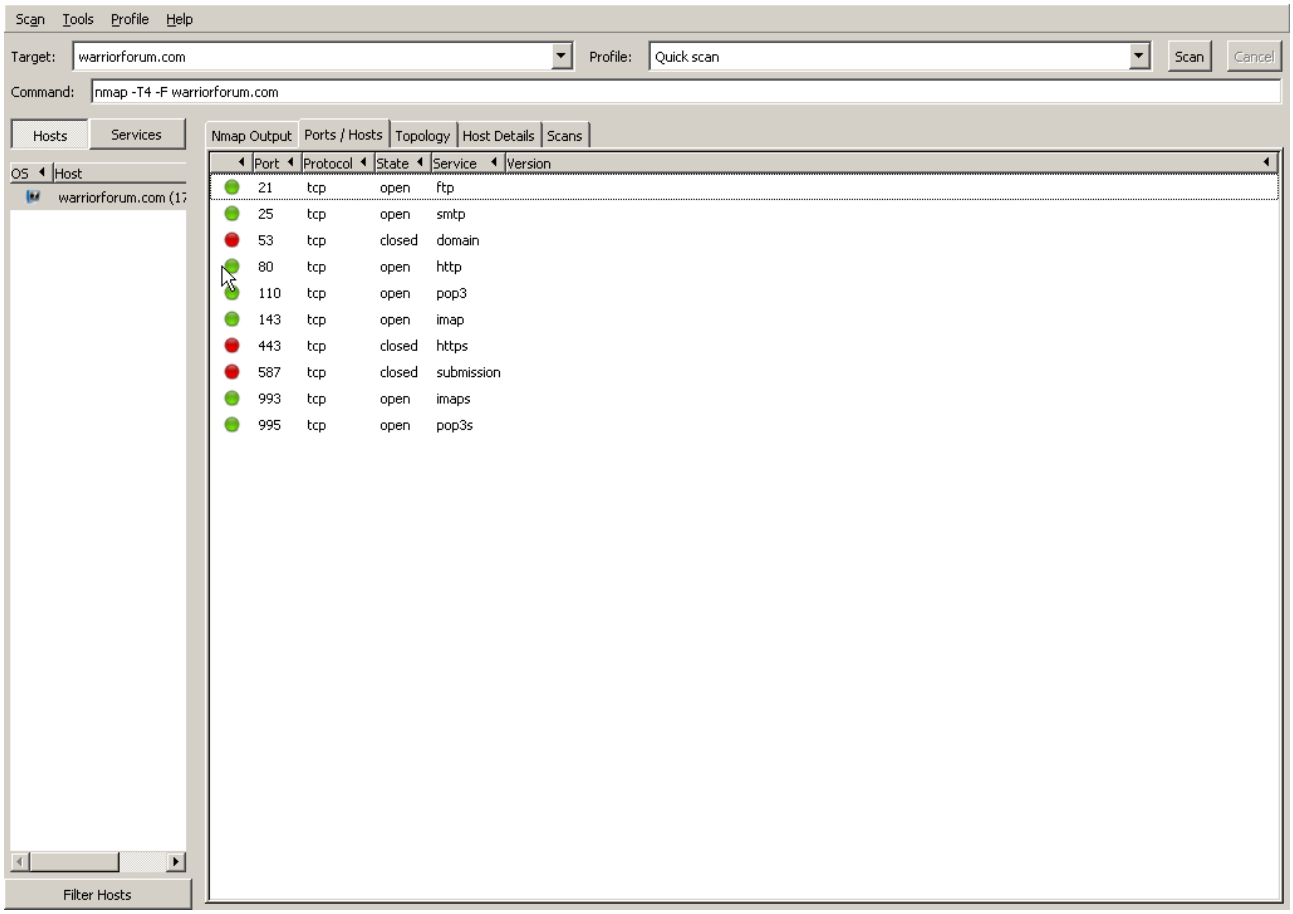
By default WordPress installs a sample post and sample page. They are called “Hello World” and “Sample Page” respectively. Once these get indexed by Google and other search engines, a hacker can do a search and will know you have a WordPress site. Delete both of them via the dashboard as (once your site is set up) they serve no useful purpose.

Checking for server-side vulnerabilities.

Making sure your WordPress site is secure is one thing, but you could be let down by things beyond your control – especially via some of the settings on your server. Unless you have your own dedicated server, these are down to your hosting company; nevertheless, you should be aware of them.

The more ports your server has open, the more opportunities a hacker has to get in. Some ports have to be open for your site to work correctly, but too many ports (or ports a hacker has opened from the server side with malice a forethought) can indicate a possible security loophole. To find out what ports are open, you can use a software program called Nmap – it's free and can be downloaded from http://nmap.org . It comes in versions for Windows, Mac and Linux.

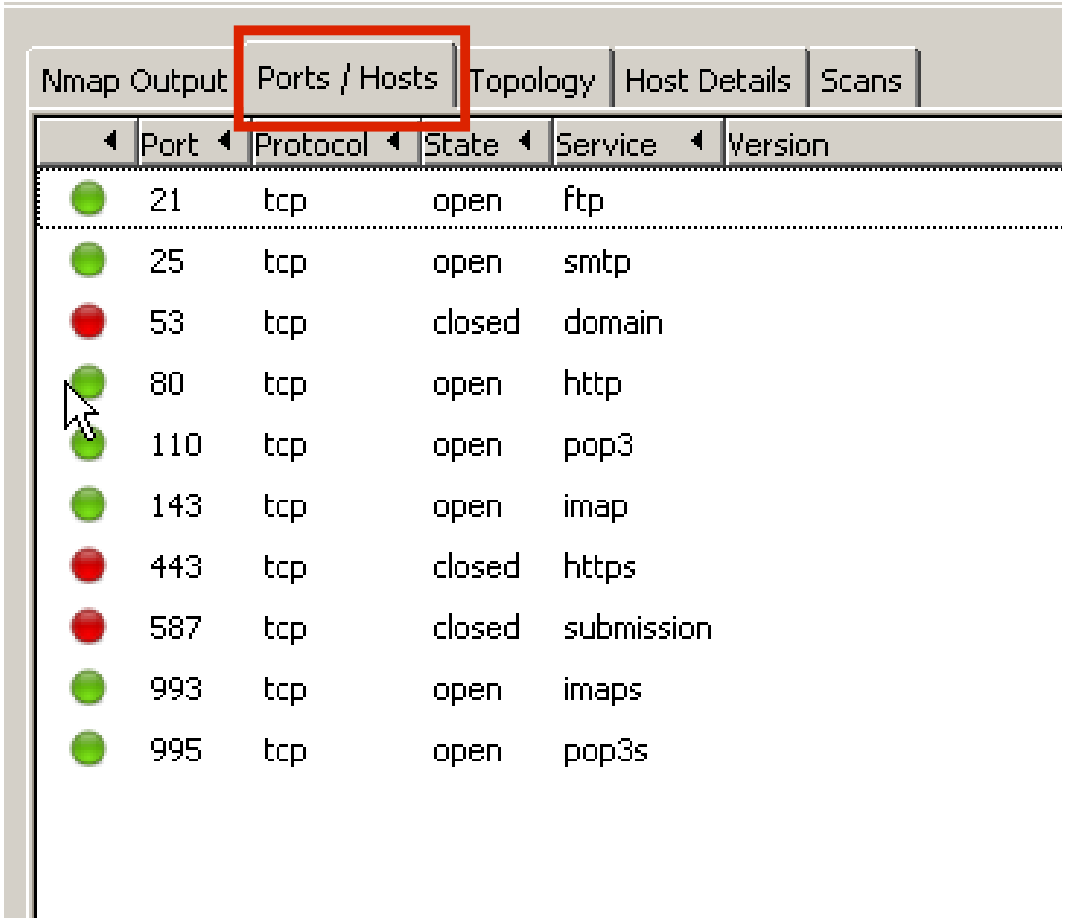
When it's downloaded and installed on your computer, you can use it to see the ports on any particular server/domain. This is what it looks like for the Warrior Forum:



Enter the domain name in the “Target” box, and select the type of scan you want from the “Profile” drop-down menu. A Quick scan should be sufficient.

It takes a few moments for the program to run depending on your Internet connection speed.

The program produces a lot of information; however, the information you're most interested in is found in the *Ports/Hosts* tab. Here it is in close-up:



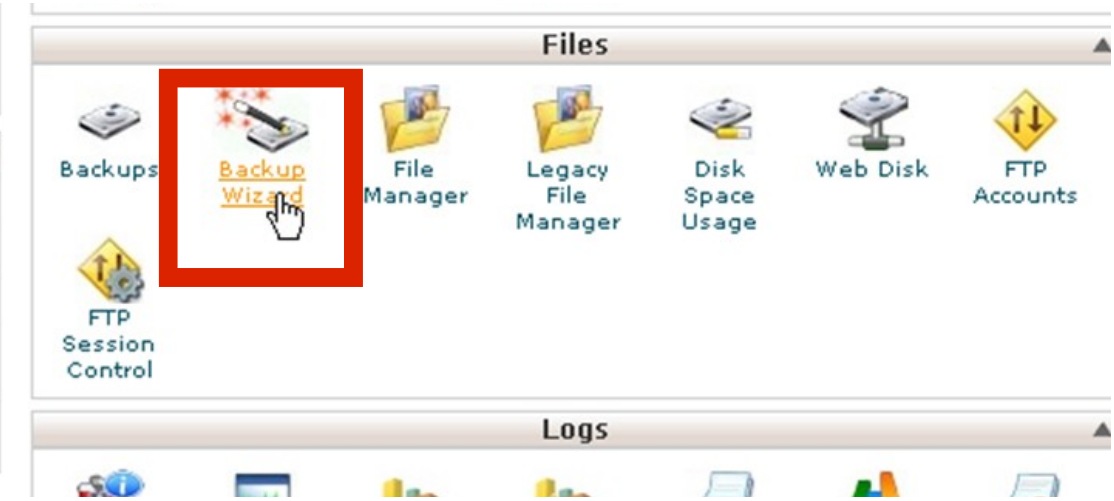
As you would expect, this is a well protected site! Ports for http and the mail servers are open, while other ports (like “submission”) have been closed.

Pay particular attention to anything in the Service column that says “unknown” as that could pinpoint a vulnerability and be worthy of further investigation.

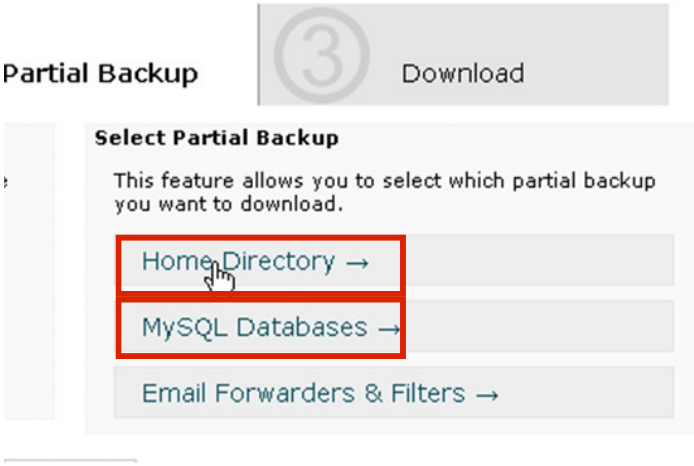
You can find out more about which ports are used for what on this Wikipedia page.

If you do find a security breach, you should bring it to the attention of your hosting company. If you're on shared hosting, there's not much more you can do. If your hosting company is unwilling or unable to tighten up server-side security, your best recourse is to take your business elsewhere to a hosting company that takes security more seriously.

And finally...



Back your site up. The best way of doing so is to do Home Directory and Database backups from within your control panel. On Cpanel, click on the "Backup Wizard" icon in the "Files" area. Other control panels have similar settings and the procedure is pretty much the same.



Select "Backup", and from the *Partial Backup* selection on the right-hand side, select "Home Directory" and "MySQL Databases". Download the zip files to your computer and back them up somewhere safe.

You should do this regularly. There are plug-ins and other programs that claim to do this for you. I've tried several, and (to be honest) they're all pretty lacklustre in my opinion. The DB backup setting in the All in One WP Security & Firewall plug-in is OK, but when you do it this way, you know it's been done properly.

Conclusion

When my sites got hacked, it was a major inconvenience. I had to spend time sorting them all out – time I should have spent building my business. While my sites were out of action, potential customers were going elsewhere.

I got of lightly this time. Thankfully I didn't have any product launches or major advertising campaigns running. Google didn't flag my site as an attack site (which would have been catastrophic) and I was able to get things up and running eventually.

What annoyed me most about the whole experience was all of this hassle could have easily been avoided, and it was all my own fault for not taking proper precautions.

You now know what I had to learn the hard way. Please put it to good use. If you haven't been hacked yet, take action today and keep the hackers out.

I wish you every success.

Warmly,

Shaun Pearce

Make Money With This Report!

Here's how …

Exclusively for War Room Members!