



# the WordPress Security checklist

*Let's make it harder for the bad guys!*

**Copyright © 2012 Ayoro SAS**

The WordPress Security Checklist is the copyright of Ayoro SAS, Cap Omega, Rond Point Benjamin Franklin, CS 39521, 34960 Montpellier, CEDEX 2, France.

All rights reserved.

No part of this checklist may be reproduced in another format or altered in any way without prior written permission from the authors.

**Disclaimer:**

Every effort has been made to ensure that this checklist contains accurate and current information. However Ayoro SAS and the authors shall not be liable for any loss or damage incurred as a result of following these instructions.

We are not responsible for how the software and services referenced or recommended in this document works.

Unless expressly mentioned, we receive no commissions or other payments from the suppliers or owners of software and services mentioned in this book.

We provide information about how to use certain third-party products, but we do not endorse or directly support third-party products and we are not responsible for the functions or reliability of such products.

WordPress is a trademark owned by The WordPress Foundation.

All other company and product names may be trademarks of the respective companies with which they are associated.

This document is protected by copyright.

You are **welcome to distribute** the document,  
however you are **not allowed to modify** it in any way.

If you did not download this document from our website  
we recommend you get the latest version from:

[www.WPSecurityChecklist.com](http://www.WPSecurityChecklist.com)

# Table of Contents

|  |           |
|--|-----------|
| <b>1 Introduction.....</b>                                   | <b>6</b>  |
| 1.1 Audience.....  | 7         |
| 1.2 Help Us Help Others.....                                 | 7         |
| 1.3 Help Us Help You.....                                    | 7         |
| 1.4 Help Us Keep The Checklist Free.....                     | 7         |
| 1.5 Help Us Make The Checklist Better.....                   | 8         |
| <b>2 Before We Get Started.....</b>                          | <b>10</b> |
| 2.1 WordPress Security Starts (And Ends) With YOU.....       | 10        |
| 2.2 Backup Your WordPress Site Now!.....                     | 10        |
| 2.2.1 WordPress Site Backup.....                             | 11        |
| 2.2.2 WordPress Site Restoration.....                        | 12        |
| 2.3 Test your site.....                                      | 14        |
| 2.4 What is Datafeedr?.....                                  | 15        |
| <b>3 Non-WordPress Security.....</b>                         | <b>17</b> |
| 3.1 Personal Computer Security.....                          | 17        |
| 3.2 Password Management.....                                 | 20        |
| 3.3 Secure FTP.....  | 22        |
| 3.4 Don't Store Passwords In Your FTP Client.....            | 24        |
| <b>4 Securing WordPress.....</b>                             | <b>27</b> |
| 4.1 WordPress Update Notifications.....                      | 27        |
| 4.2 Security Plugins.....                                    | 29        |
| 4.2.1 Login LockDown.....                                    | 29        |
| 4.2.2 Semisecure Login Reimagined.....                       | 30        |
| 4.2.3 WP Login Security.....                                 | 31        |
| 4.2.4 AntiVirus.....   | 34        |
| 4.2.5 WebsiteDefender WordPress Security.....                | 37        |
| 4.2.6 WordPress File Monitor Plus.....                       | 43        |
| 4.2.7 Update Notifications.....                              | 47        |
| 4.2.8 WordPress Firewall 2.....                              | 49        |
| 4.2.9 Block Bad Queries.....                                 | 52        |
| 4.3 Schedule Backups Of Your WordPress Site.....             | 54        |
| 4.4 Delete Unused Plugins And Themes.....                    | 55        |
| 4.5 Remove The Default Administrator User.....               | 56        |
| 4.6 Disable User Registration If Not Used .....              | 58        |
| 4.7 File Permissions.....                                    | 59        |
| 4.8 Delete The install.php File.....                         | 63        |
| 4.9 Add Empty index.php Files.....                           | 64        |
| 4.10 Move The wp-config.php File.....                        | 65        |
| 4.11 Disable File Editing From The Administration Panel..... | 67        |
| 4.12 Use Unique Keys And Salts In wp-config.php.....         | 68        |
| 4.13 Add SSL To The Admin Area.....                          | 69        |
| 4.14 Monitoring.....   | 70        |
| 4.14.1 WebsiteDefender.....                                  | 70        |
| 4.14.2 Pingdom.....  | 74        |
| 4.14.3 Change Detection.....                                 | 76        |
| 4.15 Cloudflare For Security.....                            | 78        |
| 4.16 Enable Logging And Archiving For Apache.....            | 81        |

|  |    |
|--|----|
| 4.17 Disable direct access to your database..... | 82 |
| 4.18 Intrusion Detection Systems.....            | 83 |
| 4.19 .htaccess files.....                        | 84 |
| 4.20 Securing PHP.....                           | 99 |

|                           |            |
|---------------------------|------------|
| <b>5 Rescue Plan.....</b> | <b>104</b> |
|---------------------------|------------|

# 1 Introduction

The team behind The WordPress Security Checklist run a number of WordPress sites.

And we thought we had done a good job of securing our sites.

We were wrong!

Our sites were compromised. Suspicious looking files appeared on our sites, and we had no idea how or when they managed to get on to our sites.

We cleaned up our sites, added a couple of security plugins and thought that was the end of that.

But it was only a couple of weeks before it happened again.

This time we decided good enough was not good enough. We wanted to get to the bottom of how to secure WordPress properly. After researching the topic we discovered why we had not done a good job of securing our WordPress sites in the first place.

It is very difficult to get a comprehensive and easy to understand answer to the question:

## *How do I secure my WordPress site?*

Sure, there are plenty of blog posts listing the 10 best security plugins (from two years ago) and how to secure your WordPress administration panel. In fact there is too much scattered information out there. Finding, testing and deciding which of the many bits and pieces of information are valuable is a very time consuming exercise.

Hence the birth of The WordPress Security Checklist.

This checklist is the digest of our research into the topic.

It is not perfect. It is not finished. In fact it will never be finished because WordPress continues to develop.

But it is our hope that this checklist will help you do a better job at securing your WordPress site.

And to be prepared if your site is ever broken into.

The goal of the checklist is not to explain everything in detail. It is designed to allow you to get the job done quickly.

## 1.1 Audience

The checklist is written with a non-technical reader in mind, but even WordPress experts are likely to find some very useful tips.

## 1.2 Help Us Help Others

You can do us and your network a big favor:

Help us help them.

If you like this checklist spread the word to your network.

Twit, bookmark, pinterest, Facebook like, Google +1 and link to our site!

Click [here](#) to help us. It only takes 10 seconds!



## 1.3 Help Us Help You

Sign up to our newsletter so we can keep you updated when something important happens.

We use Mailchimp to manage our mailing list, so you can safely unsubscribe at any time.

And of course we do not share your email address with anyone!

To see past emails from our newsletter click [here](#).

[Sign Up Now!](#)



## 1.4 Help Us Keep The Checklist Free

As you can imagine lots of time and money has gone into creating this checklist and the website to host it.

Please help us keep this checklist free.

Donate whatever you think the checklist is worth to you.

And, yes, even a donation of USD5 will make a difference. We are thirsty from all the writing and would really like a beer :-)

[Buy us a beer :-\)](#)



## 1.5 Help Us Make The Checklist Better

We know the checklist can be even better.

And we need your questions, thoughts and ideas.

Please [send us your feedback](#).

And if you have something to share with other users of the checklist please [give us a testimonial](#).

A simple one liner saying Thanks will be great!



*Let's make it harder for the bad guys!*



This checklist shows you how to improve the security on your WordPress site.

Many of the security checkpoints can be completed by anyone who has installed a plugin in WordPress and done basic administrative tasks.

However some of the security checkpoints are more advanced. We have done everything we can to make the checklist as easy to follow as possible.



**If you do not feel confident completing a checkpoint we recommend you skip it. Or pay someone to do it for you.**

Even if you only complete the basic security checkpoints you have greatly strengthened the security of your WordPress site.

If you are comfortable transferring files to and from your hosting account using FTP and/or you are comfortable working with the cpanel of your hosting account you should be just fine – even with the more advanced security checkpoints.

Some of the tasks described in this document can break your site if you are not careful. When editing some of the configuration files even the slightest spelling mistake could stop your site from working (until you find and correct the mistake or restore a previous version of the file).

We **strongly recommend** that you make a complete backup before you get started!

Although every possible care has been taken to ensure this checklist is 100% correct we have no control over the software and services recommended in this checklist.

**Under no circumstances can Ayoro SAS be held responsible for any losses incurred directly or indirectly as a result of following this checklist.**

## 2 Before We Get Started

### 2.1 WordPress Security Starts (And Ends) With YOU

The single most important factor in WordPress security is **YOU**.

The fact that you are reading this checklist is a great indication that you have the right mindset.

You have to be security conscious. That means you have to understand that you need to be pro-active in securing your WordPress site and that security is an ongoing activity.

Even if you implement all the security measures described here you will still be vulnerable if you do not keep your WordPress site updated. We help you stay on top of this by describing how you can receive automatic update notifications, but we cannot update your sites for you!

We continue to improve the WordPress Security Checklist, and for us to be able to notify you about changes to the checklist you need to sign up to our newsletter.

If you have not already signed up you should do this now: [Sign Up Now!](#)



### 2.2 Backup Your WordPress Site Now!

This is **extremely important**. You will be making changes to your configuration. Some of these changes could potentially break your site. Therefore you need a good backup.

If you already have a backup system in place make sure the backup includes both your database and files and that the backup is current. And that you understand the process for restoring your site if you have to.

We recommend the use of a plugin called BackWPup. It is free and allows you to backup both your files and database in one job, which can be scheduled to run automatically.

If you use and like BackWPup we recommend that you [make a donation](#) to the author.

**Tip!** Make frequent backups while you work with this checklist. When you have completed one task and everything is working fine make a backup.

## 2.2.1 WordPress Site Backup

To backup your WordPress site follow these steps:

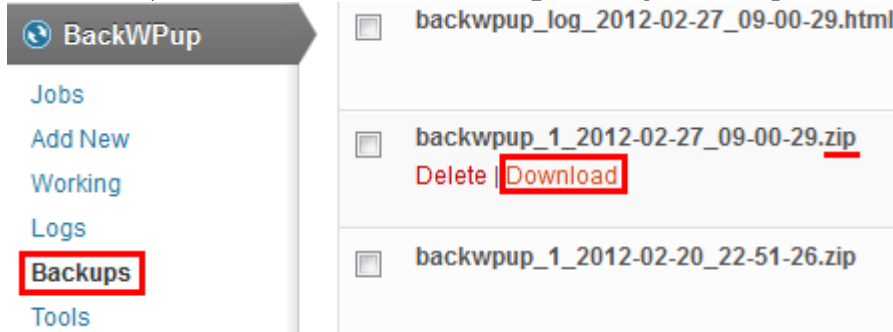
- Log in to your WordPress Administration Panel.
- Add the plugin called **BackWPup**.
- Create a new job and make sure **Database Backup** and **File Backup** are both checked.

- Make a note of the location of the backup file. BackWPup will automatically create a folder for you.

- Save your changes.
- On the **Jobs** page click **Run Now**.

| ID | Job Name    | Type  | Information                              |
|----|-------------|---|--|
| 1  | Full Backup | Database Backup<br>File Backup<br>Optimize Database Tables<br>Check Database Tables | DB Size: 41.5 MB<br>Files Size: 45.71 MB |

- When the job is done download the **zip** file to your computer.



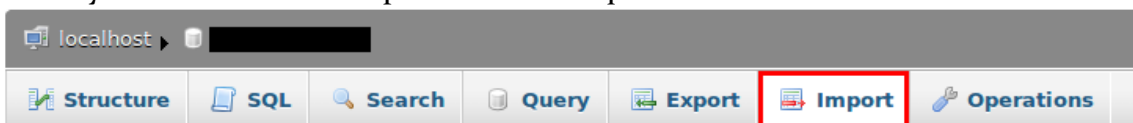
## 2.2.2 WordPress Site Restoration

If you need to restore your site follow these steps:

- Unzip the file
- Find the **xxxxx.sql** file and move it to another directory
- In your cpanel account open phpMyAdmin.  
Cpanel accounts look slightly different depending on which hosting provider you use.



- Select your database and Import the **xxxxx.sql** file.



Importing into the database " [REDACTED] "

### File to Import:

File may be compressed (gzip, zip) or uncompressed.

A compressed file's name must end in **.[format].[compression]**. Example: **.sql.zip**

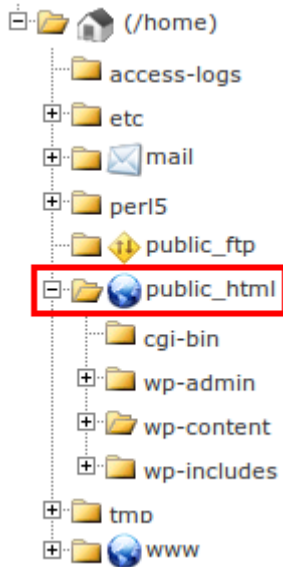
Browse your computer:   (Max: 50MiB)

Character set of the file:

- On your hosting account remove any existing files and folders from your WordPress site folder. This is typically everything **in** the public\_html, www or html folder if you only have one website on your hosting account.

**Do not remove the folder itself**, rather delete everything **inside** the folder.

**Tip!** Do this using the FileManager on your hosting account instead of your FTP client. Your FTP client would delete one file at a time and that could take some time.



- Using your FTP client transfer the remaining files from the zip file to your WordPress site folder on your hosting account.

Your site should now be in the same state as when you made the backup.

## 2.3 Test your site

Before you start working on securing your WordPress site we **strongly** recommend that you test your site to make sure everything works as expected.

- This will identify any previously unknown problems on your site.
- You will know that any new problems encountered while you work are related to changes you have just made.
- You will also have a base line to test against when you are finished securing your site so you know your site is still working as it should.

Write up a test plan detailing what you test and how.

Some ideas of what you need to test:

- That you can log in to your administration panel.
- That you can log in to your front end (if that is enabled on your site).
- That navigation works, i.e. you can click on your menus and posts and get to the pages and posts.
  - Test access to your contact, about us, privacy and other pages.
- That your URLs have not changed. Note a couple of key URLs to both page and posts and verify they do not change.
- That you can still create new pages, posts and comments to posts.
- For each plugin you use test every aspect of the functionality the plugin provides. Don't just assume that everything works!

For example:

- For a Contact Form test that the form still appears where it should, and that you can submit a request.
- For a shopping cart test that you can place items in the cart and make a purchase.
- For an image gallery test that it is still showing the correct images in the correct places, and that advanced browsing of images still works. Verify descriptions and titles are still in place.
- Verify that Google Analytics is still working.
- Generate a new sitemap and verify it is correct.
- Etc.

Keep the test plan so you can use it in the future!

## 2.4 What is Datafeedr?

Throughout the checklist you will find tips especially for Datafeedr users.



Datafeedr is a system to build affiliate stores on the WordPress platform.

We have several stores built on this platform and are very happy with the way it works.

The key advantages of Datafeedr are:

- It's very easy to build good looking stores – even for non-technical people.
- You can build stores combining products from several different affiliate networks without any additional work.

In fact **you never have to touch a datafeed** yourself!

- Products and prices are updated **automatically**!

This is extremely important. Once you've built your store it will automatically update itself.

- Very, very good support via their support forum.

Click to see a 5 minute video introduction to Datafeedr (affiliate link):





Click to visit a sample store:

## Best Contact Lens Prices

Find the Best Contact Lens Prices Online

[Home](#) [About](#) [Deals](#)

[Contact Lenses](#) » [Browse by Brand](#) » [Acuvue](#)

### Acuvue Contact Lenses

Products per page:   Sort by:

Pages: [1](#) [2](#) > >>

#### [1-DAY ACUVUE TruEye 90 Pack Contact Lenses](#)



[Find The Best Price](#)

#### [ACUVUE Advance Plus \(24-Pack\) Contact Lenses](#)



[Find The Best Price](#)

#### [1-DAY ACUVUE MOIST 90 Pack Contact Lenses](#)



[Find The Best Price](#)

#### [ACUVUE Oasys for Presbyopia Contact Lenses](#)



[Find The Best Price](#)

#### [ACUVUE Oasys for Astigmatism Contact Lenses](#)



[Find The Best Price](#)

#### [ACUVUE Advance for Astigmatism Contact Lenses](#)



[Find The Best Price](#)

#### Search Contact Lenses

#### Browse Contact Lenses

- [Browse by Brand](#)
  - [Acuvue](#)
  - [Air Optix](#)
  - [Avaira](#)
  - [Biofinity](#)
  - [Biomedics](#)
  - [Boston](#)
  - [Clearsight](#)
  - [CSI](#)
  - [Dailies](#)
  - [Durasoft](#)
  - [Expressions](#)
  - [Frequency](#)
  - [Focus](#)
  - [FreshLook](#)
  - [Hydrasoft](#)
  - [O2 Optix](#)
  - [Optima](#)
  - [Preference](#)
  - [Proclear](#)
  - [Purevision](#)
  - [SofLens](#)
  - [Vertex](#)
- [Browse by Type](#)

#### Latest Deals

**CONTACTS**America.com

\$100 New Wearer Rebate with purchase of 8 boxes or more of 1-DAY ACUVUE® MOIST®. Valid thru 06/30/12  
Begins: 01/01/2012 12:00 AM  
Ends: 06/30/2012 12:00 AM



## 3 Non-WordPress Security

This section describes security measures that take place outside of your WordPress site.

### 3.1 Personal Computer Security

#### 3.1.1.1 What You Need To Do

Ensure that the computers you use to log in to your blog:

- have a good anti-virus and malware scanner installed
- have a firewall installed
- have the latest Operating System updates

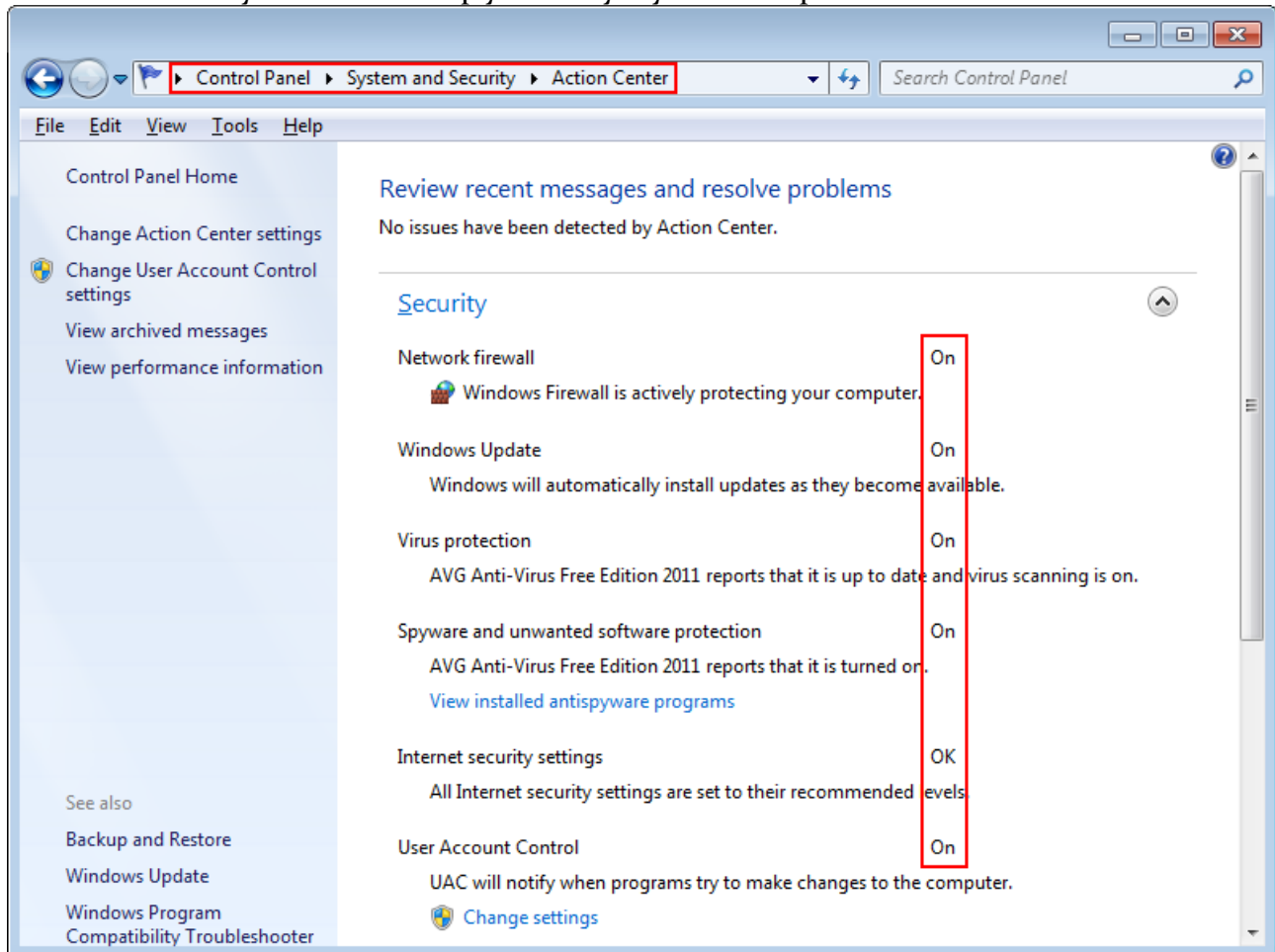
This is especially true on Windows computers but also applies to Mac and Linux.

#### 3.1.1.2 Why This Point Is Important

If your computer has been infected with malware your WordPress login details can be stolen and sent silently to a hacker, who now has full access to your WordPress site.

### 3.1.1.3 How You Complete This Security Checkpoint

Windows Security Center will help you verify if you have a problem.



If you are using a computer that is not your own remember to check that it is fully updated before using it to log in to your website.

Be cautious about using unknown computers (e.g. in Internet Cafés) to access the administration area of your WordPress site.

Regularly (once per month) run two or more of the free online scanners recommended below.

### 3.1.1.4 Recommendation

If you already have a good anti-virus program then keep it.

Otherwise we recommend [AVG Free](#).

Windows standard firewall should be fine. If you are looking for an alternative we recommend [ZoneAlarm Free Version](#).

**Note!** No single anti-virus program will protect you from all vira and malware. We recommend that you regularly scan your computer using free online scanners. This is especially true if you suspect your computer might have been infected.

- [The Secunia Online Software Inspector](#)
- [Trend Micro Housecall](#)
- [Bitdefender Online Scanner](#)
- [Panda ActiveScan](#)

### 3.1.1.5 Further Resources

- [PCMag – The Best Free Antivirus for 2012.](#)
- [MakeUseOf – The Three Best Free Firewalls for Windows.](#)

## 3.2 Password Management

### 3.2.1.1 What You Need To Do

Always use strong passwords and use different passwords for different accounts.

### 3.2.1.2 Why This Point Is Important

If you use dictionary words as passwords they are too easy to guess. Robots on the internet try brute force attacks where they try to log on to your site with easy to guess user name and password combinations. If successful the winning combination is transmitted to the hacker, who now has full access to your site. So if your user name is “admin” and your password is “password” you have a problem.

Another trap is to reuse user names and passwords across different sites to make them easier to remember. If your combination is discovered it can be abused on other sites you are registered on.

### 3.2.1.3 How You Complete This Security Checkpoint

With the ever increasing number of user names and passwords we have to remember it is best to use a password manager.

A good password manager should help you:

- Generate strong passwords.
- Remember your user names and passwords for you.
- Automatically log you in to the websites you have accounts on.

### 3.2.1.4 Recommendation

If you already are using a Password Manager and you are happy with it don't change anything.

Otherwise we recommend [LastPass](#). It slots into your internet browser and on top of remembering all your user names and passwords also automatically logs you in. It works on most browsers and Windows, Mac and Linux. And by the way... it's free... unless you need the mobile version in which case it is very cheap.

Drawback: It only works through your browser, so you have to manually copy your password into other programs, like your FTP client for example.

### 3.2.1.5 Further Resources

Another popular Password Manager is [RoboForm](#).

[Keepass](#) is an open source password manager.

## 3.3 Secure FTP

### 3.3.1.1 What You Need To Do

If you access your hosting account with an FTP client you need to use Secure FTP.

### 3.3.1.2 Why This Point Is Important

If you connect via regular FTP the user name and password to your hosting account is sent over the Internet in clear text. So if someone is 'listening in' on your conversation they can steal your login information. This is especially risky when you are connected via public wifi.

### 3.3.1.3 How You Complete This Security Checkpoint

In your FTP client you have to change the connection method to Secure FTP.

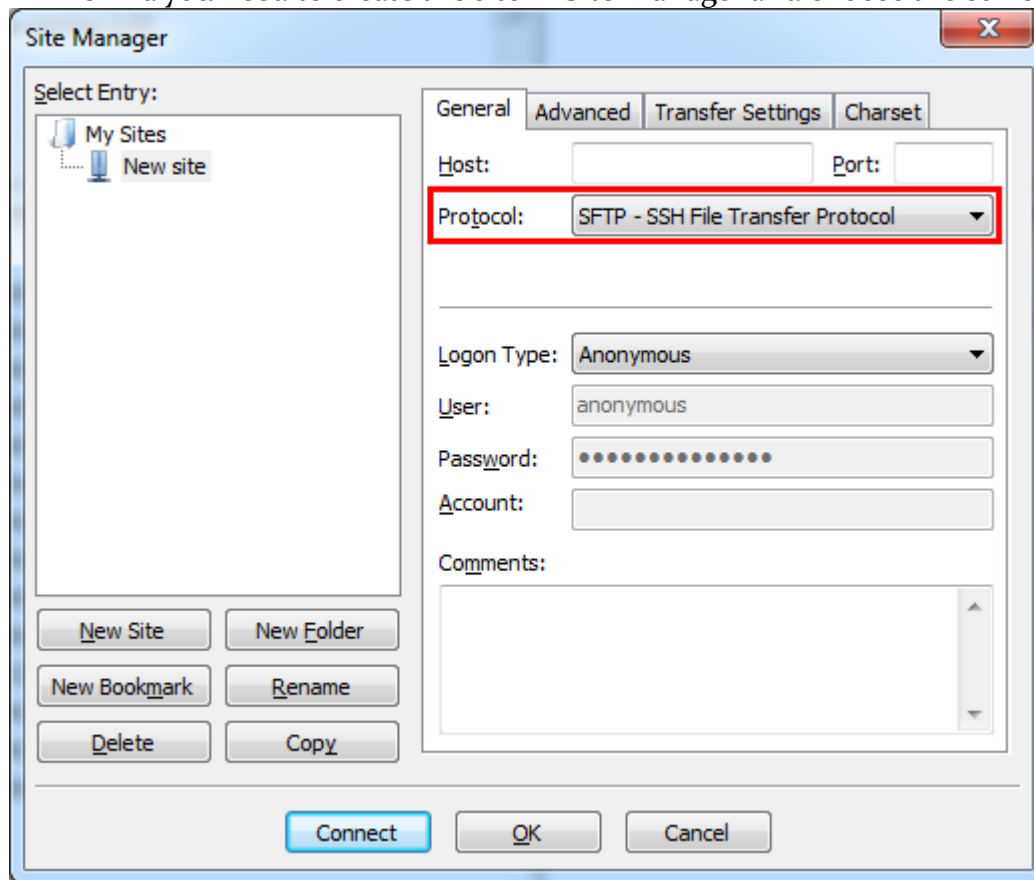
Secure FTP can go by many different names. Most often it is one of the following:

- SFTP
- FTPS
- FTPS-SSL

The type of Secure FTP you can use depends on what your hosting provider has enabled. If necessary ask them. If they have no way of connecting via Secure FTP you should probably consider changing hosting provider!

The way to change the connection method depends on your FTP client.

In FileZilla you need to create the site in Site Manager and choose the correct protocol.



### 3.3.1.4 Further Resources

- [FileZilla Web Site](#).
- [WordPress training on FileZilla](#).
- [Article on using SFTP from Make Tech Easier](#).

## 3.4 Don't Store Passwords In Your FTP Client

### 3.4.1.1 What You Need To Do

If you access your hosting account with an FTP client you should not store your password in the FTP client!

### 3.4.1.2 Why This Point Is Important

A common way for hackers to gain unrestricted access to hosting accounts is via malware programs that steal login information from FTP clients. A small program is installed on your computer. It will look for an FTP client, and if it finds one it will steal all the login details and send them back to the hacker. Without you even knowing it happened!

So you should not store the passwords to your hosting accounts in your FTP client.

You can enable all the other security mechanisms detailed in this checklist, but if a hacker gets your login details from your FTP client they will have unrestricted access until you change your password.

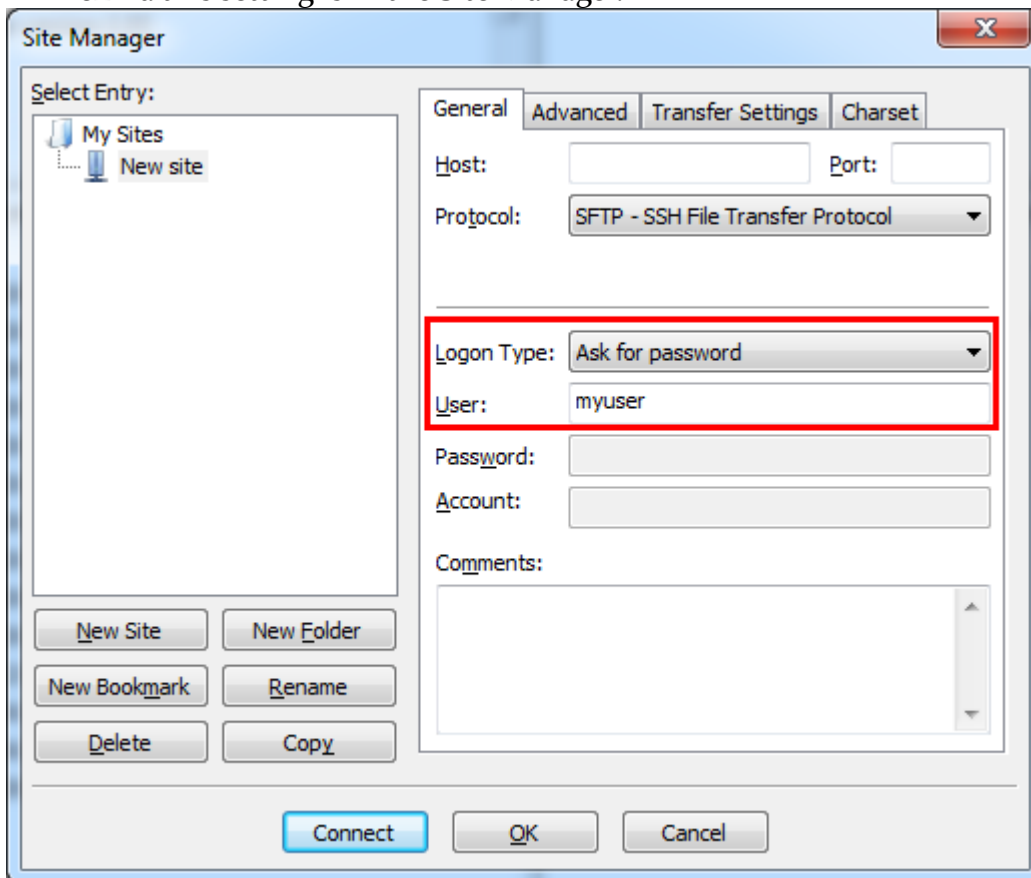
You will need to copy and paste the password from your password manager every time you login... but it is much better than giving hackers access to your hosting account.

### 3.4.1.3 How You Complete This Security Checkpoint

In your FTP client change the settings so you will be prompted for the password when you connect to your hosting account.



In FileZilla this setting is in the Site Manager.



# Easy-Email is the solution to your email problems!

## Problem: Too many email addresses

How many email addresses do you check every day?

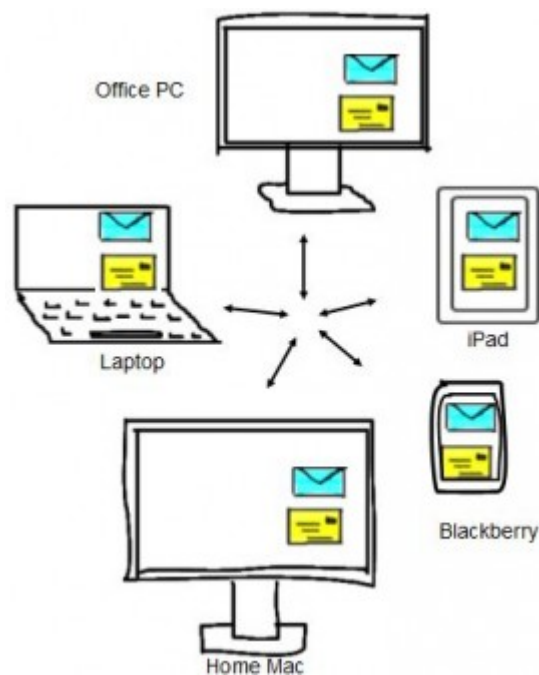


## Solution: Aggregate your email!

With Easy-Email you get **all** your email in **one place** with **one login**!

## Problem: Too many computers

Can't get **all** your email on **all** your computers?



## Solution: Synchronize your email!

With Easy-Email your email is **automatically synchronized** on **all** your **computers and devices**!

## What is Easy-Email?

[Click to watch our introduction video.](#)

## With Easy-Email you can:

- have your email come to you, so you only need to check one account!
- have all your email on your netbook **and** your main computer **and** your phone!
  - read your email offline: look up that hotel address in the taxi!
- read, respond to and organize your email while you're on a plane!
- send email from any of your email addresses from your iPhone!
  - check your office email (Outlook for instance) from home or anywhere else on the planet!

[www.Easy-Email.net](http://www.Easy-Email.net)

# 4 Securing WordPress

This section details how you keep your WordPress site safe and secure and how to be informed if there is suspicion of an intrusion.

## 4.1 WordPress Update Notifications

### 4.1.1.1 What You Need To Do

Subscribe to the official WordPress email notifications.

These notifications are only for WordPress core updates – not plugins. We will discuss how to receive notifications about updates for plugins in [Update Notifications](#) on page 47.

### 4.1.1.2 Why This Point Is Important

It is extremely important that you keep your WordPress site updated.

Once an update is released you need to apply it to your site as soon as possible to close any security holes identified in the release.

#### WordPress 3.3.2 (and WordPress 3.4 Beta 3)

Posted April 20, 2012 by [Andrew Nacin](#). Filed under [Development](#), [Releases](#), [Security](#).

[WordPress 3.3.2](#) is available now and is a security update for all previous versions.

### 4.1.1.3 How You Complete This Security Checkpoint

Follow these steps:

- Go to the [WordPress download page](#)

- Enter your email and click **Join** to subscribe.

#### Release Notification

We've got a handy mailing list that we send a friendly message to whenever there's a new stable release for you to enjoy.

 [WordPress Releases RSS](#)

### Important!

Tips for a successful WordPress update:

- Always backup before you update.
- Never update a live site directly.  
Create a test site and update your test site first.  
If everything works fine you can proceed to update your live site.
- Minor updates do not add functionality so are less likely to break things.  
E.g. 3.3.2 → 3.3.3 is a minor update.
- Other updates change WordPress functionality so more testing is required.  
E.g. 3.3.2 → 3.4 or 3.4 → 4.0
- Do not modify WordPress core files manually.  
Changes will be overwritten on the next update.
- If you use WordPress.org hosted themes use child themes to modify the theme behavior instead of modifying the theme code. Again any changes to the theme code will be overwritten if the theme is updated.



To create a test site:

- Create a test site on a subdomain to your main domain name.  
If your main domain is [www.mywebsite.com](#) you can create a test site on [mytest.mywebsite.com](#) – provided your hosting plan allows you to run more than one site.  
**Note!** If you leave your test site on your hosting account you need to keep this site updated at all times as well. Don't just update your main site.
- Or use [xamp](#) to install and run a test site locally on your own computer.

## 4.2 Security Plugins

The section describes the WordPress plugins we recommend you use to improve your security.

Are we missing your favorite WordPress security plugin? [Click here to let us know!](#)

### 4.2.1 Login LockDown

Plugin Page: <http://wordpress.org/extend/plugins/login-lockdown/>

#### 4.2.1.1 What You Need To Do

Ensure that brute force attempts to guess your user name and password are stopped.

#### 4.2.1.2 Why This Point Is Important

Login LockDown keeps an eye out for failed login attempts. If someone tries to log in too many times with a wrong user name/password combination that IP address will be blocked automatically.

This an effective way to stop user name/password guessing.

#### 4.2.1.3 How You Complete This Security Checkpoint

Add and Activate the plugin. The default values are fine for most WordPress sites.

Settings

General

Writing

Reading

Discussion

Media

Privacy

Permalinks

Login LockDown

Login LockDown Options

Max Login Retries

3

Retry Time Period Restriction (minutes)

5

Lockout Length (minutes)

60

Lockout Invalid Usernames?

☐ Yes
☒ No

Mask Login Errors?

☐ Yes
☒ No

Update Settings

Currently Locked Out

No current IP blocks locked out.

Release Selected

#### 4.2.1.4 Recommendation

In case you ever lock yourself out you can disable this plugin by renaming (or removing) the plugin folder **wp-content/plugins/login-lockdown**. Or wait an hour to login again.

#### 4.2.2 Semisecure Login Reimagined

Plugin Page: <http://wordpress.org/extend/plugins/semisecure-login-reimagined/>

**Note!** This plugin is no longer under active development. However it still works and the author is encouraging the community to take over the development in the future.

##### 4.2.2.1 What You Need To Do

Ensure your login details are not sent over the Internet in clear text.

##### 4.2.2.2 Why This Point Is Important

You might be unable to use SSL for your administration panel (see [Add SSL To The Admin Area](#) on page 69). If this is the case the Semisecure Login Reimagined plugin is the next best thing.

It will automatically encrypt the password in the Internet Browser when you login. This will make it much harder for a third party to steal your password.

Otherwise your password will be sent over the Internet in clear text.

##### 4.2.2.3 How You Complete This Security Checkpoint

Add and Activate the plugin. The default values are fine for most WordPress sites.

## 4.2.3 WP Login Security

Plugin Page: <http://wordpress.org/extend/plugins/wp-login-security/>

### 4.2.3.1 What You Need To Do

Add an extra layer of security to your login process.

### 4.2.3.2 Why This Point Is Important

WP Login Security intelligently adds another layer of security to the login process.

The plugin keeps track of the IP addresses used by administrators. If an administrator tries to login from an unknown IP address an activation link is emailed to the registered email address of the administrator. Until the activation link is clicked the administration panel is blocked.

Even if someone steals your WordPress user name and password they will be unable to login unless they also have access to your email.

### 4.2.3.3 How You Complete This Security Checkpoint

Add and Activate the plugin.

- The default settings are fine.

**Settings** WP Login Security

WP Login Security allows each user to maintain a whitelist of IP addresses allowed to login to the site.

Enable WP Login Security? ☒

Notify Both Blog Admin & User? ☐

[Save Changes](#)

**Whitelisted IP Addresses**

| Username | IP Address | Date Activated   |
|----------|------------|------------------|
| admin    | [REDACTED] | 04-21-2012 10:30 |

**Outstanding IP Activations**

| Username | IP Address | Request Date | Activation Key |
|----------|------------|--------------|----------------|
| Username | IP Address | Request Date | Activation Key |

When you login from a new IP address you will see a blank screen. An email with the activation link will be sent to your email address:

```
To: info@mywebsite.com
Subject: [My Website] WP Login Security Alert

Someone has logged in with the below information from an IP we haven't
seen before.

User: admin
IP: xxx.xxx.xxx.xxx
URL: http://www.mywebsite.com/wp-admin/

To authorize this IP address, please click the following link:
http://www.mywebsite.com/wp-login.php?
action=registerip&wpls_ipkey=d41d8cd98fasdfas98837498ecf8427e
```

#### 4.2.3.4 Recommendation

In case you experience difficulties logging in you can always disable this plugin by renaming (or removing) the plugin folder **wp-content/plugins/wp-login-security**.



#### 4.2.3.5 Further Resources

There are three other two factor authentication plugins you might want to consider.

**Note!** We have not tested these plugins.

##### 4.2.3.5.1 Second factor

Plugin Page: <http://wordpress.org/extend/plugins/second-factor/>

Second factor adds another layer to the login process making it more secure.

The first factor is your user name and password. This plugin will email a one time code to the users email address. This code has to be entered before the login is complete.

Even if someone gets your user name and password they will be unable to login unless they also have access to your email.

##### 4.2.3.5.2 Google Authenticator

Plugin Page: <http://wordpress.org/extend/plugins/google-authenticator/>

The Google Authenticator plugin for WordPress gives you two-factor authentication using the Google Authenticator app for Android/iPhone/Blackberry.

You may already have the Google Authenticator app installed on your smartphone, using it for two-factor authentication on your Gmail or Google Apps account.

The two-factor authentication requirement can be enabled on a per-user basis. You could enable it for your administrator account, but log in as usual with less privileged accounts.

##### 4.2.3.5.3 Duo Two-Factor Authentication

Plugin Page: <http://wordpress.org/extend/plugins/duo-wordpress/>

This plugin enables Duo Security's two-factor authentication for WordPress logins.

Duo provides simple two-factor authentication as a service via:

- Phone callback
- SMS-delivered one-time passcodes
- Duo mobile app to generate one-time passcodes
- Duo mobile app for smartphone push authentication
- Duo hardware token to generate one-time passcodes

## 4.2.4 AntiVirus

Plugin Page: <http://WordPress.org/extend/plugins/antivirus/>

### 4.2.4.1 What You Need To Do

Monitor your theme files for malware and virus.

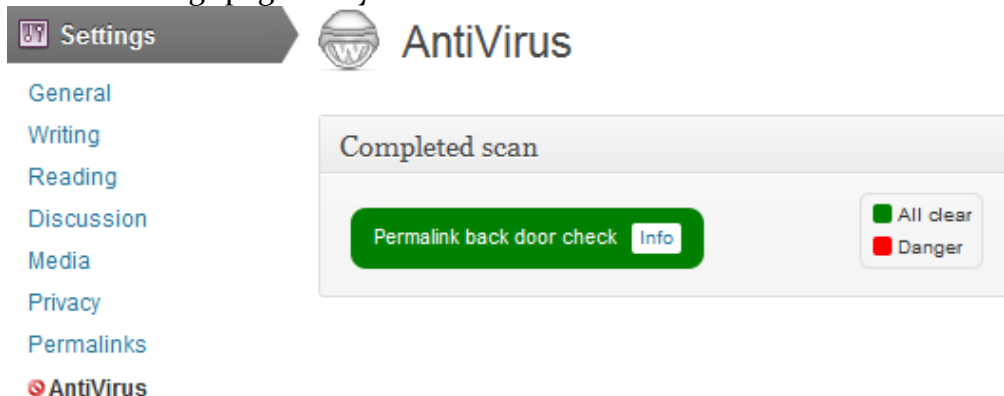
### 4.2.4.2 Why This Point Is Important

This plugin will scan your themes on a daily basis and send you an email if it finds suspicious code. It is a strong indication of an attack on your site if your theme files change when you have not updated your site.

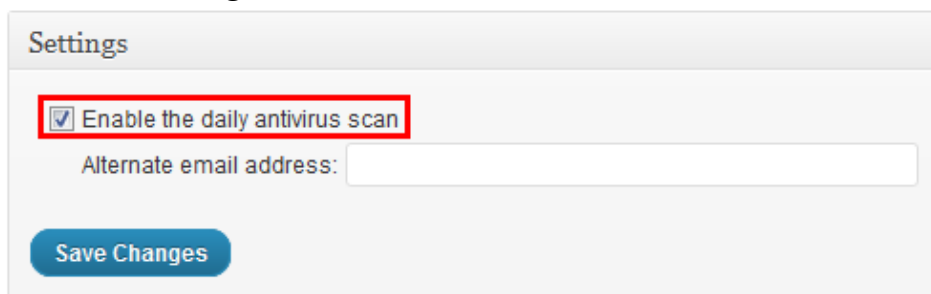
### 4.2.4.3 How You Complete This Security Checkpoint

Add and Activate the plugin.

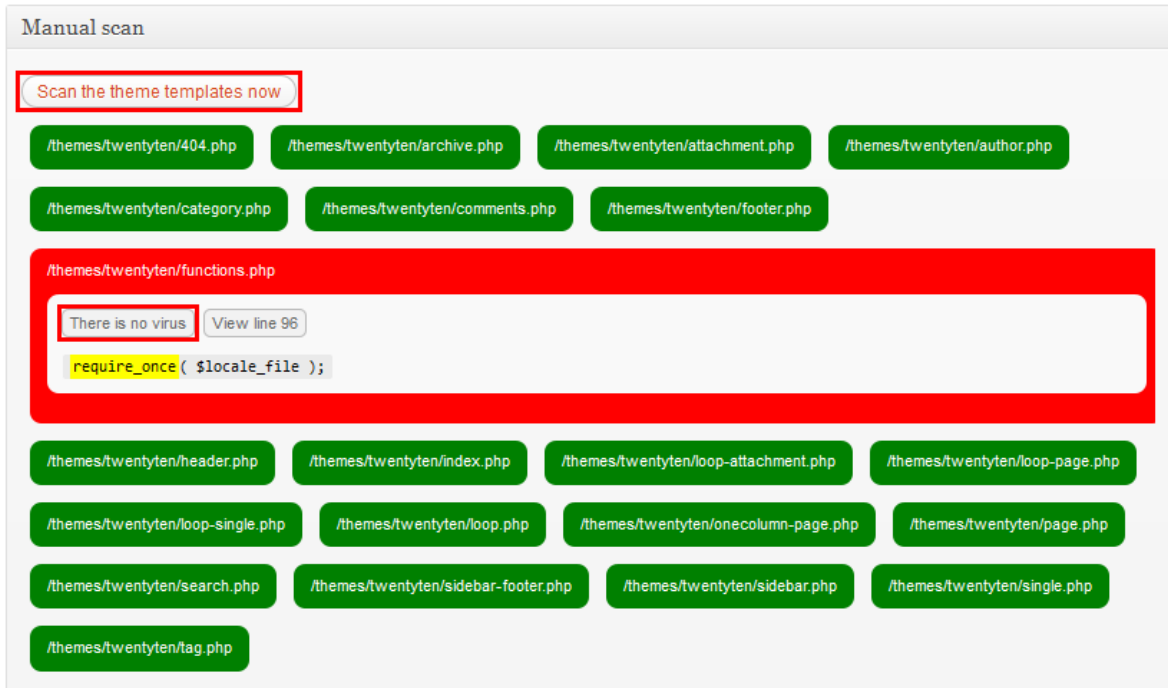
- On the settings page verify that the Permalink back door check is clean.



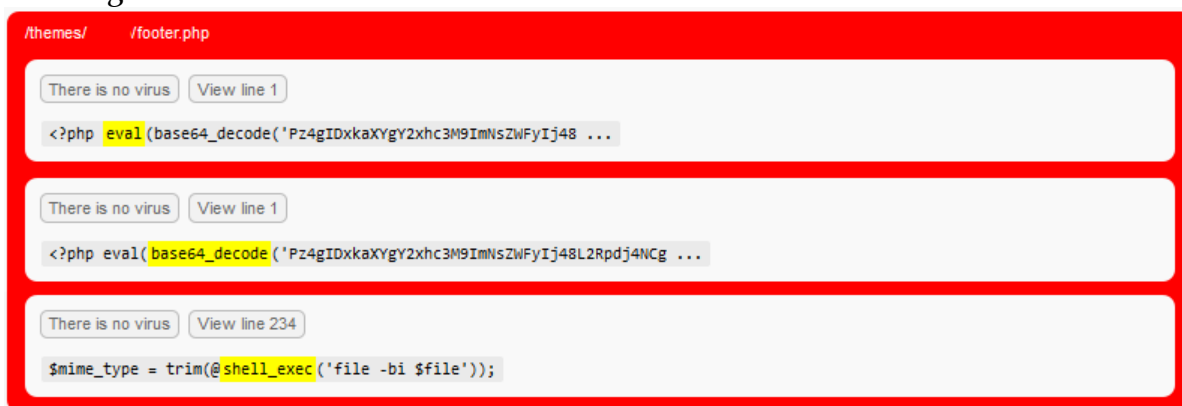
- Enable the daily scanning.  
Click **Save Changes**.



- Do a manual scan of the theme.  
**Note!** You will more often than not get false positives. The scanner flags **require\_once**, **include** and other code, which is used both in legitimate code but also in malicious code.
- Scanning results for the official WordPress TwentyTen theme with one false positive. Here you can click **There is no virus** to accept the code.



- Scanning results for a theme **with suspicious** code.  
A theme should **not** use **eval**, **base64\_decode** or **shell\_exec**.  
In this case either the theme has been infected or it contained suspicious code from the beginning.  
If you suspect the theme has been infected you can re-install the theme and run the check again.



#### 4.2.4.4 Recommendation

We recommend that you do not use themes (free or paid for) that contain base64 encoded code.

#### 4.2.4.5 Further Resources

Excellent articles on free WordPress themes:

- [Why You Should Never Search For Free WordPress Themes in Google or Anywhere Else](#)
- [When is a Free WordPress Theme Really Free? Some Thoughts and Some Places to Find Them](#)

## 4.2.5 WebsiteDefender WordPress Security

Plugin Page: <http://wordpress.org/extend/plugins/websitedefender-wordpress-security/>

**Note!** There are two other plugins with similar names in the plugin directory: Secure WordPress and WP Security Scan. They are both older versions with less functionality than WebsiteDefender WordPress Security.

### 4.2.5.1 What You Need To Do

Strengthen your basic WordPress security.

### 4.2.5.2 Why This Point Is Important

The WebsiteDefender Wordpress Security plug-in tightens up the security of your WordPress site by:

- Removing error-information on the login-page.
- Checking for the existence of index.php in the wp-content, wp-content/plugins, wp-content/themes and wp-content/uploads directories to prevent directory listings.
- Removing the wp-version, except in admin-area.
- Removing Really Simple Discovery meta tag.
- Removing Windows Live Writer meta tag.
- Removing core update information for non-admins.
- Removing plugin-update information for non-admins.
- Removing theme-update information for non-admins.
- Hiding wp-version in backend-dashboard for non-admins.
- Removing version on URLs from scripts and stylesheets only on frontend.
- Providing various security information after scanning your WordPress site.
- Providing file permissions security checks.
- Providing a tool for changing the database prefix.
- Turning off database error reporting (if enabled).
- Turning off PHP error reporting.

### 4.2.5.3 How You Complete This Security Checkpoint

Follow these steps to complete the security checkpoint.

- Add and Activate the plugin.

- On the main window for WSD Security you have the option to create an online account for the WSD scanning service. We will do this in [WebsiteDefender](#) on page 70.

The screenshot shows the WebsiteDefender WordPress Security interface. On the left is a sidebar menu with options: Plugins, Users, Tools, Settings, WSD Security (highlighted), WSD Security, Scan Reports, Database, Password, Options, and About WSD. The main content area has a header 'WebsiteDefender WordPress Security' and a sub-header 'WebsiteDefender'. Below this is a login form with the text 'Login here if you already have a WSD account:'. The form includes fields for 'Email:' (containing 'admin@mysite.com') and 'Password:', and a 'Login' button.

- On the Scan Reports page WSD will check for common security problems.
- In the **Wordpress Scan Report** section all but the last item should have a green tick.


The screenshot shows the 'Wordpress Scan Report' page. The sidebar menu is the same as in the previous screenshot, with 'WSD Security' highlighted and 'Scan Reports' selected. The main content area has a header 'Wordpress Scan Report'. Below this is a list of 15 items, each with a green checkmark indicating a successful check. The items are:
 

- You have the latest version of Wordpress. ✓
- Your database prefix is not wp\_. ✓
- The Wordpress version is hidden for all users but administrators. ✓
- Database errors are not displayed. ✓
- PHP errors are not displayed. ✓
- Startup errors are not displayed. ✓
- User admin was not found. ✓
- The .htaccess file was found in the wp-admin directory. ✓
- The index.php file was found in the wp-content directory. ✓
- The index.php file was found in the plugins directory. ✓
- The index.php file was found in the themes directory. ✓
- The index.php file was found in the uploads directory. ✓
- The readme.html file was found in the root directory ([view file](#)). It is very important to either delete this file or make it inaccessible from your browser as it displays your Wordpress version! ⚠


- If you do not have green ticks you should do this:
  - You **do not** have the latest version of Wordpress:  
Upgrade to the latest version.

- Your database prefix **should not be** wp\_.  
Use WSD to change the table prefix (see [Changing The Database Table Prefix Using WSD](#) on page 41).
- The Wordpress version is hidden for all users but administrators.  
This is done by WSD so should always be a green tick.
- Database errors are not displayed.  
This is done by WSD so should always be a green tick.
- PHP errors **are** being displayed.  
Follow the instructions in [Securing PHP](#) on page 99.
- Startup errors **are** being displayed.  
Follow the instructions in [Securing PHP](#) on page 99.
- User admin **was** found.  
Follow the instructions in [Remove The Default Administrator User](#) on page 56.
- The .htaccess file was **not** found in the wp-admin directory.  
Follow the instructions in [WordPress wp-admin folder](#) on page 91.
- The index.php file was **not** found in the wp-content directory.  
Follow the instructions in [Add Empty index.php Files](#) on page 64.
- The index.php file was **not** found in the plugins directory.  
Follow the instructions in [Add Empty index.php Files](#) on page 64.
- The index.php file was **not** found in the themes directory.  
Follow the instructions in [Add Empty index.php Files](#) on page 64.
- The index.php file was **not** found in the uploads directory.  
Follow the instructions in [Add Empty index.php Files](#) on page 64.
- The readme.html file was found in the root directory.  
This point will not get a green tick. Make sure you have followed the instructions in [# Stop access to sensitive files](#) on page 88.
- In the **File Scan Report** files and folders with permissions different from the WSD recommendation are flagged.  
A higher permission number indicates more relaxed permissions. If permissions are more relaxed than they have to be you could have a security problem.  
Or in other words: A lower number than recommended is fine... a higher number is not good.  
For more details on permissions see [File Permissions](#) on page 59.


- Below is an example of the File Scan Report.  
The first three flags (.htaccess and wp-config.php) are because we have more restrictive permissions than the WSD recommendation (i.e. a lower number).  
The last flag (readme.html) indicates we have more relaxed permissions (i.e. a higher number). In the section [# Stop access to sensitive files](#) on page 88 we will restrict access to this particular file.

 File Scan Report


| Name                 | Path                                 | Current permissions | Suggested permissions |
|----------------------|--------------------------------------|---------------------|-----------------------|
| ✓ root directory     | /home/public_html/                   | 0755                | 0755                  |
| ✓ wp-admin           | /home/public_html/wp-admin           | 0755                | 0755                  |
| ✓ wp-content         | /home/public_html/wp-content         | 0755                | 0755                  |
| ✓ wp-includes        | /home/public_html/wp-includes        | 0755                | 0755                  |
| ⚠ .htaccess          | /home/public_html/.htaccess          | 0444                | 0644                  |
| ⚠ wp-config.php      | /home/public_html/wp-config.php      | 0640                | 0644                  |
| ✓ wp-admin/index.php | /home/public_html/wp-admin/index.php | 0644                | 0644                  |
| ⚠ wp-admin/.htaccess | /home/public_html/wp-admin/.htaccess | 0444                | 0644                  |
| ⚠ readme.html        | /home/public_html/readme.html        | 0644                | 0440                  |

 Our suggested permissions are still secure but more permissive in order to not break some servers' setups. If your existent file permissions are more restrictive, ex: 0750 instead of the suggested 0755 then you have no reason to change it to the suggested 0755 permissions.

- The File Scan Report from a local test server.  
You can see **this WordPress is not secure**, but as it is only accessible from a local computer and not from the Internet it does not matter.

 File Scan Report

| Name                 | Path   | Current permissions | Suggested permissions |
|----------------------|--|---------------------|-----------------------|
| ⚠ root directory     | C:\xampp\htdocs\wordpress/                   | 0777                | 0755                  |
| ⚠ wp-admin           | C:\xampp\htdocs\wordpress/wp-admin           | 0777                | 0755                  |
| ⚠ wp-content         | C:\xampp\htdocs\wordpress/wp-content         | 0777                | 0755                  |
| ⚠ wp-includes        | C:\xampp\htdocs\wordpress/wp-includes        | 0777                | 0755                  |
| ⚠ .htaccess          | C:\xampp\htdocs\wordpress/.htaccess          | 0666                | 0644                  |
| ⚠ wp-config.php      | C:\xampp\htdocs\wordpress/wp-config.php      | 0666                | 0644                  |
| ⚠ wp-admin/index.php | C:\xampp\htdocs\wordpress/wp-admin/index.php | 0666                | 0644                  |
| ⚠ wp-admin/.htaccess | C:\xampp\htdocs\wordpress/wp-admin/.htaccess | 0666                | 0644                  |
| ⚠ readme.html        | C:\xampp\htdocs\wordpress/readme.html        | 0666                | 0440                  |

 Our suggested permissions are still secure but more permissive in order to not break some servers' setups. If your existent file permissions are more restrictive, ex: 0750 instead of the suggested 0755 then you have no reason to change it to the suggested 0755 permissions.

- In case you have problems with your file permissions see [File Permissions](#) on page 59.
- If you have changed the database prefix from wp\_ you are finished with this plugin for now. Otherwise continue with the next step.



#### 4.2.5.3.1 Changing The Database Table Prefix Using WSD

WSD makes it easy to change the database table prefix.

Follow these instructions.

- **Important!** Make a backup of your WordPress site using BackWPup.

The screenshot shows the BackWPup interface. On the left, there's a sidebar with 'BackWPup' selected. Below it are links for 'Jobs', 'Add New', 'Working', and 'Logs'. The main area displays a table of backup jobs.

| ID | Job Name   | Type                           | Information                              |
|----|--|--------------------------------|--|
| 1  | Daily<br><a href="#">Edit</a>   <a href="#">Copy</a>   <a href="#">Export</a>   <a href="#">Delete</a>   <a href="#">Run Now</a> | Database Backup<br>File Backup | DB Size: 2.28 MB<br>Files Size: 25.33 MB |


- In WSD click on the **Database** tab.  
Verify your wp-config.php is **writable** and you have **ALTER** rights to the database.  
Enter your own unique table prefix. Do not use **myprefix\_**.  
Click **Start Renaming**.

The screenshot shows the 'Change Database Prefix' screen in WSD Security. The left sidebar has 'Database' selected. The main content area has a heading 'Change Database Prefix' and a sub-heading 'Change your database table prefix to avoid zero-day SQL Injection attacks.' Below this, it says 'Before running this script:' followed by three bullet points: 'Backup your database.', 'The wp-config.php file must be **writable** (Yes)', and 'The database user you're using to connect to database must have ALTER rights (Yes)'. There are two informational boxes: one about file permissions and another about database user rights. At the bottom, there's a text input field for the current table prefix (showing 'myprefix\_') and a 'Start Renaming' button.

- The tasks completes.
- Test that your site still works.
- The Scan Report now has a green tick for the table prefix.

The screenshot shows the 'Wordpress Scan Report' screen in WSD Security. The left sidebar has 'Database' selected. The main content area has a heading 'Wordpress Scan Report' and a list of three items, each with a green checkmark: 'You have the latest version of Wordpress.', 'Your database prefix is not wp\_ .', and 'The Wordpress version is hidden for all users but administrators.'

- Notice the warning about the database user having too many rights. This is actually not true, so you can safely disregard that warning. The default rights given to the database user are in fact required. See [this forum post](#) for more information.

 The database user used to access the WordPress Database has too many rights . Limit the user's rights to increase your Website's Security.

- As detailed in [this post](#) you could lower the database privileges during normal operation of your blog. But note that you will most likely have to change the privileges before you upgrade WordPress or the plugins or if you install new plugins.

#### 4.2.5.4 Further Resources

- [Article about Backdoor Scripts.](#)
- [Article about the WebsiteDefender WordPress Security plugin.](#)

## 4.2.6 WordPress File Monitor Plus

Plugin Page: <http://wordpress.org/extend/plugins/wordpress-file-monitor-plus/>

### 4.2.6.1 What You Need To Do

Install and enable the plugin.

### 4.2.6.2 Why This Point Is Important

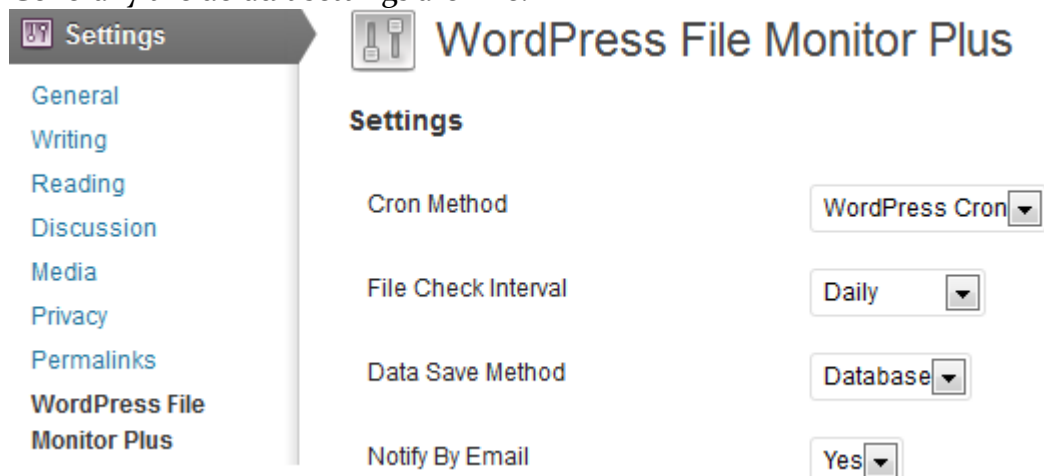
This plugin will monitor all the files in your WordPress site. If someone breaks into your site they will most likely add files to your site. These extra files can act as backdoors, which can potentially allow hackers to execute files from their own servers. These files can hijack your traffic, place unwanted ads or links on your pages and place malware on your visitors computers.

With the File Monitor you will be notified by email if anything in the file system changes. This will allow you to quickly clean up a hacking attempt, because you know exactly which files have been modified and when the hack occurred. Best option is to restore a recent backup from before the hacking attempt took place to wipe out any changes the hackers might have made to the database as well. See [Rescue Plan](#) on page 104 for more information on how to recover from a hack.

### 4.2.6.3 How You Complete This Security Checkpoint

Follow these steps:

- Add and Activate the plugin.
- Generally the default settings are fine.



- Files and directories you might want to exclude from the file monitor.

Dirs/Files To Ignore

```
*w3tc/*
*w3-total-cache-config.php
*/sitemap.xml
*/sitemap.xml.gz
*/error_log
*/store/*
```

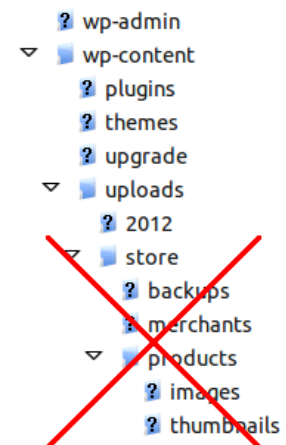
- Your caching plugins working directory.  
For W3 Total Cache this would be the **wp-content\w3tc folder**.  
For WP Super Cache this would be the **wp-content\cache folder**.
- Your caching plugins configuration files if they are updated often by the plugin.  
Example **w3-total-cache-config.php**.
- Your sitemap files.
- The error\_log file.

#### Datafeedr Tip! ([What is Datafeedr?](#))

If you are using Datafeedr you should be storing your product image files locally (for performance reasons).

You can choose to exclude the store folder from the file scan.  
This will stop the file monitor from sending emails every time a product image is added to the local folder.

Once your products images have been fully downloaded we recommend that you include the store folder in your file scan again.



- We recommend that you leave the **File Extensions Scan** disabled. Using this option you can exclude certain file types, image files for example, from the scan. Often malicious code is disguised as graphics files, so you should monitor all files.

File Extensions Scan

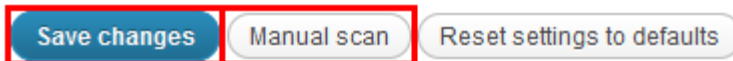
Disabled

File Extensions

.jpg|.jpeg|.jpe|.gif|.png|.bmp|.tif|.tiff|.ico

Separate extensions with | character.

- Click **Save changes** then **Manual scan**.



- Verify that you receive an email.

[Clear admin alert](#)

Files Changed: 35  
Files Added: 10  
Files Removed: 0

**Files Changed:**

| File   | New Filesize | Old Filesize | New Modified          | Old Modified              |
|--|--------------|--------------|-----------------------|---------------------------|
| /wp-content/plugins/all-in-one-seo-pack/aioseop.class.php            | 99 kB        | 98 kB        | April 1, 2012 @ 13:20 | December 14, 2011 @ 01:01 |
| /wp-content/plugins/all-in-one-seo-pack/aioseop_options.php          |              |              | April 1, 2012 @ 13:20 | December 14, 2011 @ 01:01 |
| /wp-content/plugins/all-in-one-seo-pack/all_in_one_seo_pack-bg_BG.mo | 18 kB        | 18 kB        | April 1, 2012 @ 13:20 | December 14, 2011 @ 01:01 |

**Tip!** When you update a plugin you will receive an alert. Sometimes quite a few files have been updated. The list of file changes is ordered by directory, so you can quickly check that only the plugin files have been updated by verifying the first and last files in the list.

Beginning of the list:

Files Changed: 98  
Files Added: 0  
Files Removed: 0

**Files Changed:**

| File   | New Filesize | Old Filesize |
|--|--------------|--------------|
| /wp-content/plugins/contact-form-7/admin/admin.php               | 10 kB        | 10 kB        |
| /wp-content/plugins/contact-form-7/admin/edit.php                | 6 kB         | 6 kB         |
| /wp-content/plugins/contact-form-7/admin/images/dropdown.gif     | 67 B         | 67 B         |
| /wp-content/plugins/contact-form-7/admin/images/menu-icon.png    | 606 B        | 606 B        |
| /wp-content/plugins/contact-form-7/admin/images/screen-icon.png  | 1 kB         | 1 kB         |
| /wp-content/plugins/contact-form-7/admin/includes/meta-boxes.php | 4 kB         | 4 kB         |

End of the list:

|  |       |       |
|--|-------|-------|
| /wp-content/plugins/contact-form-7/styles.css            | 887 B | 887 B |
| /wp-content/plugins/contact-form-7/uninstall.php         | 456 B | 456 B |
| /wp-content/plugins/contact-form-7/wp-contact-form-7.php | 2 kB  | 2 kB  |

[Clear admin alert](#)

All updated files are in the plugin directory in this example.

**Tip!** Run a manual scan before you update plugins. This will ensure no files have been added before you do the update. Once the update has completed run another manual scan and accept the changes. This way you are certain that all the changes you accept are directly related to your upgrade.

#### 4.2.6.4 Recommendation

We recommend that you use a cron job to run the File Monitor scan.



### WordPress File Monitor Plus

#### Settings

Cron  
Method

Other Cron

▼

*Cron Command:*

```
wget -q "http://www.██████████.com/index.php?sc_wpfpmp_scan=1&sc_wpfpmp_key="
```

If you use the built in WordPress Cron the File Monitor scan will only run if there are visitors to your site. And you cannot control the time the job runs.

If you setup a cron job on your hosting account or dedicated server to run the File Monitor scan you know that it will run every day and at what time it will run. This could be important in determining which backup to use in case you ever need to restore your site. This is discussed further in [Rescue Plan](#) on page 104.

The command you need to run in the cron job is given below the Cron Method setting. Ask your hosting company how to set up the cron job.

## 4.2.7 Update Notifications

Plugin Page: <http://wordpress.org/extend/plugins/update-notifications/>

### 4.2.7.1 What You Need To Do

You need to keep your WordPress site updated at all times.

### 4.2.7.2 Why This Point Is Important

As soon as an update is released anyone can see *why* the update was released.

The sooner you update the sooner you close any security holes that are now public knowledge.

### 4.2.7.3 How You Complete This Security Checkpoint

You need to be informed about updates to core WordPress, plugins and themes.

The Update Notifications plugin has given the best results in our testing. If you wish you can also try some of the other plugins mentioned in the Further Resources section.

- Add and Activate the plugin.  
The default settings will work for most sites, but you can modify the configuration in case you want more email addresses to be notified for example.

**Settings**

- General
- Writing
- Reading
- Discussion
- Media
- Privacy
- Permalinks
- Update Notifications**

### Update Notifications

Added Email

*To enter multiple email addresses separated by a comma*

Checks for updates every  Hours

Disable notifications for these items

- ☐ Wordpress Core
- ☐ Plugins
- ☐ Themes

**Save**

- **Important!** If you use a WordPress Plugin which is not in the WordPress Plugin Directory make sure they have a mailing list for updates and get on the list.

#### 4.2.7.4 Further Resources

Other plugins in the same category:

- [Mail on update](#)
- [Update Notifier](#)
- [WP Updates Notifier](#)
- [Upgrade Notification by Email](#)
- [Plugin Update Notification](#)
- [Update Notifications](#)



## 4.2.8 WordPress Firewall 2

Plugin Page: <http://wordpress.org/extend/plugins/wordpress-firewall-2/>

### 4.2.8.1 What You Need To Do

Protect your WordPress site from malicious requests.

### 4.2.8.2 Why This Point Is Important

A commonly used way for hackers to try to gain access to your site is by embedding malicious code in requests to your site.

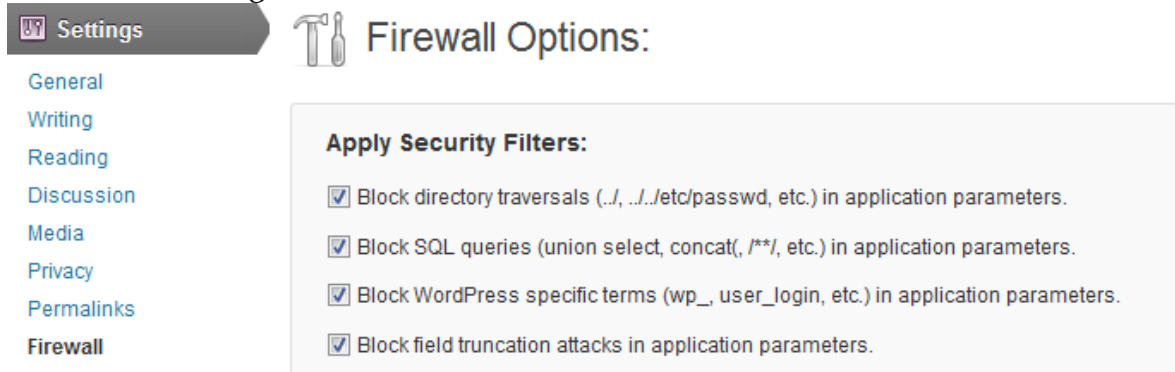
As an example a hacker might embed code to add an administrator user to the database in a request from the internet. This is also known as a SQL Injection Attack.

The WordPress Firewall 2 plugin will stop these types of attacks.

### 4.2.8.3 How You Complete This Security Checkpoint

Follow these steps:

- Add and Activate the plugin.
- The default settings will work for most sites.



- **Optional:** Turn off email notifications.  
Whenever the Firewall stops an attack it will send you an email with details of the attack.

**WordPress Firewall has detected and blocked a potential attack!**

**Web Page:** www.██████████.com/tinymce/plugins/imagemanager/classes/CorePlugin.php?basepath=../../../../../../../../../../../../proc/self/environ%00  
Warning: URL may contain dangerous content!

**Offending IP:** 184.105.174.107 [ [Get IP location](#) ]

**Offending Parameter:** **basepath = ../../../../../../../../../../../../../../proc/self/environ\0**

This may be a "Directory Traversal Attack."

[Click here](#) for more information on this type of attack.

If you repeatedly get attacked from a particular IP address you can block access to your site from this IP address. Hackers usually disguise their real IP addresses and run automated attacks using other peoples computers, so in our opinion blocking IP addresses has little value.

See how to block an IP address from your site: [Block IP Address](#).

If you do not want to receive notification emails from your Firewall enter a blank address and click **Set Email**.

**Email:**

Enter an email address for attack reports:

Note: Leave this setting blank to disable emails.

Email type: ☒ html ☐ text

Suppress similar attack warning emails: ☐ On ☒ Off

Set Email

- **Optional:** Whitelist your own IP address.  
If you edit certain settings or files on your site the Firewall might think it's an attack. For example this can happen if you edit your theme files via the WordPress administration panel. If that happens you will be redirected to the home page when you try to save the file.  
You can disable the Firewall temporarily. Or you can whitelist your own IP address. This only works well if you have a fixed IP address.

**Whitelisted IPs:**

Enter IP(s) that are whitelisted — and not subject to security rules.

Note: Set field(s) to blank to disable IP whitelist. Your current IP is: XXXXXXXXXX

Set Whitelisted IPs

- **Tip!** If you use the W3Total Cache plugin we recommend you add **w3tc\_referrer** as a whitelist form variable.

**Whitelisted Pages:**

Enter page and/or form variables that are whitelisted — and not subject to security rules:

| Page: | Form Variable: |
|-------|----------------|
| *     | w3tc_referrer  |
|       |                |

Note: Set field(s) to blank to disable page whitelist.  
Note: Use \*'s for wildcard characters.

Set Whitelisted Pages

#### 4.2.8.4 Recommendation

We recommend that you use both WordPress Firewall 2 and [Block Bad Queries](#) as they protect against different types of attacks.

#### 4.2.8.5 Further Resources

- [Wikipedia article on SQL injection attacks.](#)

## 4.2.9 Block Bad Queries

Plugin Page: <http://wordpress.org/extend/plugins/block-bad-queries/>

### 4.2.9.1 What You Need To Do

Protect your WordPress site from malicious requests.

### 4.2.9.2 Why This Point Is Important

A commonly used way for hackers to try to gain access to your site is by embedding malicious code in requests to your site.

The Block Bad Queries plugin checks for excessively long request strings (i.e., greater than 255 characters), as well as the presence of either "eval(" or "base64" in the request URI.

### 4.2.9.3 How You Complete This Security Checkpoint

Add and Activate the plugin.

There are no options for this plugin.

### 4.2.9.4 Recommendation

We recommend that you use both [WordPress Firewall 2](#) and Block Bad Queries as they protect against different types of attacks.



Like what you see?

[Buy us a beer!](#)

Or give us a [testimonial!](#)

Thanks ;-)

## 4.3 Schedule Backups Of Your WordPress Site

### 4.3.1.1 What You Need To Do

You need to backup the complete WordPress site on a regular basis.

You also need to store the backups safely outside of your hosting account.

### 4.3.1.2 Why This Point Is Important

No site will ever be 100% secure.

If your site is compromised you need to be able to restore it quickly. The quickest and safest way to recover after your site has been compromised is by restoring a good backup.

You need to keep a number of backups in case the attack on your site is discovered after some time.

### 4.3.1.3 How You Complete This Security Checkpoint

Please [click here](#) to see the full set of instructions on our website.

## 4.4 Delete Unused Plugins And Themes

### 4.4.1.1 What You Need To Do

Remove anything you do not use from WordPress, e.g. disabled plugins and themes.

### 4.4.1.2 Why This Point Is Important

All files in your WordPress root folder are accessible from the Internet regardless of whether you use them or not. Even if you disable a plugin the files are still there and they are accessible from the Internet.

This is a potential security risk, as you may not pay attention to upgrading plugins and themes you are not using.

### 4.4.1.3 How You Complete This Security Checkpoint

Remove any plugins and themes you are not actively using. Disabling is not enough.

## 4.5 Remove The Default Administrator User

### 4.5.1.1 What You Need To Do

Remove the default administrator user.

### 4.5.1.2 Why This Point Is Important

Most WordPress sites are installed with wp\_ as the database table prefix and admin as the default administrator user. This makes it too easy for hackers to break into your site.

Example: You have just installed your new WordPress site using the default settings. A hacker wants to gain access to your WordPress administration panel.

He goes to [www.mysite.com/wp-admin](http://www.mysite.com/wp-admin) enters admin as user name and now all he has to do is to guess the password to get in. And all hackers have password guessing programs in their toolbox...

So in effect you have left the front door half way open!

Example: A hacker tries to reset your administrator password by sending SQL (database) code to your website. He tries to update the password for the user with id = 1. This would be the first user that was created and by default this is your admin user. So even if you installed WordPress with another user name than admin, you can still improve your security.

Renaming the default admin user is not enough!

### 4.5.1.3 How You Complete This Security Checkpoint

Follow these steps:

- In the WordPress administration panel go to **Users** → **Add New**.
- Fill in the details.



- Make sure you use a strong password and select **Administrator** as the Role.

Users

All Users

**Add New**

Your Profile

Tools

Settings

WSD Security

Collapse menu

Password (twice, required)

Strong

Hint: The password should be at least 8 characters long. To make it strong, use a mix of upper and lower case letters, numbers and symbols like ! " ? \$ % ^ & ).

Send Password?

☐ Send this password to the new user by email.

Role

**Administrator**

Add New User

- Log out and log back in using your new administrator user.
- Go to **Users** → **All Users**.
- You can now delete the default administrator user.

| <input type="checkbox"/> | Username | Name                 | E-mail                 | Role          | Posts |
|--------------------------|----------|----------------------|------------------------|---------------|-------|
| <input type="checkbox"/> | admin    |                      | admin@mysite.com       | Administrator | 1     |
|                          |          | <a href="#">Edit</a> | <a href="#">Delete</a> |               |       |

- Attribute any existing posts to your new administrator user.

What should be done with posts and links owned by this user?

☐ Delete all posts and links.

☒ Attribute all posts and links to: **newadmin**

- If your new administrator user is going to post articles make sure the Display name is different to the Username. You find this setting the profile of the user. This will make it harder for a hacker to guess your username.

## 4.6 Disable User Registration If Not Used

### 4.6.1.1 What You Need To Do

Make sure you do not allow users to register as members on your site unless you need them to. If you do need users to be able to register make sure you give them the minimum user role required.

### 4.6.1.2 Why This Point Is Important

Any access you give to your site from the Internet is a potential security risk. Unless needed turn it off.

### 4.6.1.3 How You Complete This Security Checkpoint

Follow these steps:

- In your WordPress administration panel go to **Settings** → **General**.
- Make sure **Anyone can register** is unchecked.

Membership

☐ Anyone can register

New User Default Role

Subscriber ▼

## 4.7 File Permissions

### 4.7.1.1 What You Need To Do

Ensure that your file and directory permissions are as strict as possible, in particular for special files.

### 4.7.1.2 Why This Point Is Important

Permissions on files and directories determine who is allowed to read, write and execute files in your hosting account. If your permissions are too open you might allow visitors from the Internet to read files with sensitive information or perhaps even to update them.

### 4.7.1.3 How You Complete This Security Checkpoint

Permission settings will vary from host to host and between shared hosting and dedicated hosting. If you are in doubt ask your host about their specific requirements.

Normally you will only have to verify the permission settings for a few special files (described below).

#### 4.7.1.3.1 Overview

You can specify permissions for three different user roles:

- The **user** who owns the file/directory.
- The **group** that owns the file/directory.
- **Everybody** else.

There are three types of permissions:

- **Read** (4) – the right to see the file/directory and read the contents – numerical value 4.
- **Write** (2) – the right to modify the contents (i.e. change a file or create a new file in a directory) – numerical value 2.
- **eXecute** (1) – the right to execute the file as a program – numerical value 1.

The permissions can be expressed as a three digit number like this:

- First digit is the sum of the **user's** rights. If the user can Read (4), Write (2) and eXecute (1) the first digit will be  $4 + 2 + 1 = 7$ .

- Second digit is the sum of the **group**'s rights. If the group can Read (4) and eXecute (1) but not Write (2) the second digit will be  $4 + 1 = 5$ .
- The third digit is the sum of the rights for **everybody** else (also called world). If everybody else can Read (4) and eXecute (1) the third digit will be  $4 + 1 = 5$ .

Or expressed as one number the permission would be 755.

|                 |             |             |
|-----------------|-------------|-------------|
| 7               | 5           | 5           |
| User            | Group       | World       |
| $4 + 2 + 1 = 7$ | $4 + 1 = 5$ | $4 + 1 = 5$ |

Common permissions are:

777: Owner, Group and World all have Read, Write and eXecute.

755: Owner can Read, Write and eXecute, Group and World can Read and eXecute.

644: Owner can Read and Write, Group and World can Read.

444: All can Read only.

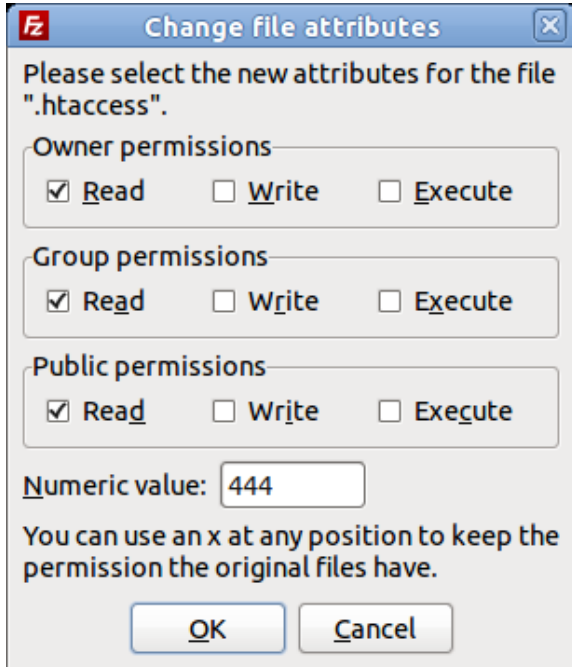
600: Owner can Read and Write, Group and World have no permissions.

400: Owner can Read, Group and World have no permissions.

The default recommendation for WordPress is that files have permission settings of 644 and folders have 755.

#### 4.7.1.3.2 Setting Permissions For A File Or Directory

In FileZilla right click on the file for directory, then enter the number or click the permissions tick boxes.



#### 4.7.1.3.3 wp-config.php file

We recommend you start with 400, and test if your site works. Some plugins might need to update this file in which case you can try 440, 600 and 640 in that order until you find a combination that works. If a plugin needs to write to wp-config.php only on installation consider lowering the permissions after the plugin has been installed.

#### 4.7.1.3.4 uploads directory

If at all possible avoid using 777 permissions.

First try 755, then 766 and finally 777 if nothing else works. Also if the permission is only needed for a short period of time make sure you change back to a more restrictive set of permissions after the need has been satisfied.

If you have a plugin that requires 777 permissions consider contacting the plugin developer and request a workaround that does not require 777.

#### 4.7.1.3.5 .htaccess files

First try 400, and test if your site works. Then try 440, 444, 600, 640 and finally 644 in that order until you find a combination that works.

#### 4.7.1.3.6 php.ini file(s)

First try 400, and test if your site works. Then try 440, 444, 600, 640 and finally 644 in that order until you find a combination that works.

#### 4.7.1.4 Further Resources

- [WordPress codex on file permissions.](#)
- [The dangers of 777.](#)
- For more info on file permissions watch [this video](#) from 20:41 until about 25 minutes.

## 4.8 Delete The install.php File

### 4.8.1.1 What You Need To Do

If the install.php file exists in your wp-admin folder delete it.

### 4.8.1.2 Why This Point Is Important

Depending how you installed WordPress the installation file might still exist in your wp-admin folder.

This potentially poses a security risk.

### 4.8.1.3 How You Complete This Security Checkpoint

Go to your wp-admin folder.

If the install.php file exists simply delete it.

### 4.8.1.4 Further Resources

- [Excellent article on the security risk of leaving the install.php file on the server.](#)

## 4.9 Add Empty index.php Files

### 4.9.1.1 What You Need To Do

Place empty index.php files within the following folders:

- wp-includes
- wp-content
- wp-content/plugins
- wp-content/themes
- wp-content/uploads

### 4.9.1.2 Why This Point Is Important

Placing an empty index.php in a folder will prevent directory browsing, which could give a potential hacker information about your themes, plugins etc.

### 4.9.1.3 How You Complete This Security Checkpoint

WordPress will automatically create most of these files for you, but not all of them.

In each of the folders mentioned above make sure you have an index.php file with this content:

```
<?php
// Silence is golden.
?>
```



## 4.10 Move The wp-config.php File

### 4.10.1.1 What You Need To Do

If your WordPress site is installed in the public\_html folder you can move the wp-config.php file one level up. This is typically the case if you only have one website on your hosting account.

### 4.10.1.2 Why This Point Is Important

The wp-config.php file is a very important configuration file. It contains a lot of sensitive information about your WordPress site, like your database information for example.

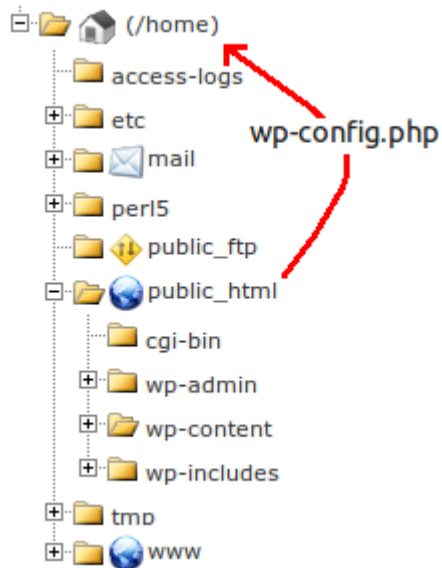
WordPress will automatically look for this file in the folder above the WordPress root folder if it does not exist in the root folder.

Moving this file out of the public\_html folder means the file will not be accessible from the Internet.

If your WordPress site is installed in a sub-folder below the public\_html folder you can only make use of this trick for one WordPress site. And please be aware that if you have another site installed in the public\_html folder the wp-config.php file might be visible on that site.

#### 4.10.1.3 How You Complete This Security Checkpoint

- Move the file from your WordPress root folder to the folder above.



- Test that your site still works.

## 4.11 Disable File Editing From The Administration Panel

### 4.11.1.1 What You Need To Do

Disable file editing from the WordPress Administration panel.

### 4.11.1.2 Why This Point Is Important

If a hacker gets access to your administration panel he can modify your theme and plugin files.

By disabling file editing you ensure files can only be updated using SFTP or the file manager on your hosting account.

### 4.11.1.3 How You Complete This Security Checkpoint

Add the following code to the top of your wp-config.php file.

```
/* BEGIN WordPress Security Checklist Addition: Disable Editor */  
define('DISALLOW_FILE_EDIT', true);  
/* BEGIN WordPress Security Checklist Addition: Disable Editor */
```

### 4.11.1.4 Recommendation

If a third party person needs to work on your site he might need access to edit files directly from the administration panel.

In that case comment out the line above, but only for as long as the need exists.

## 4.12 Use Unique Keys And Salts In wp-config.php

### 4.12.1.1 What You Need To Do

Your wp-config.php file contains a number of keys and salts that make it harder to hack your site. You need to ensure that these keys are unique.

### 4.12.1.2 Why This Point Is Important

Adding unique keys and salts strengthens your security by making it harder to hack your site and crack your passwords.

### 4.12.1.3 How You Complete This Security Checkpoint

Depending on how you installed WordPress you might already have unique keys and salts.

To verify open your wp-config.php file.

If you see the code below you need to update the keys and salts:

```
define('AUTH_KEY',          'put your unique phrase here');  
define('SECURE_AUTH_KEY',  'put your unique phrase here');  
define('LOGGED_IN_KEY',    'put your unique phrase here');  
define('NONCE_KEY',        'put your unique phrase here');  
define('AUTH_SALT',        'put your unique phrase here');  
define('SECURE_AUTH_SALT', 'put your unique phrase here');  
define('LOGGED_IN_SALT',   'put your unique phrase here');  
define('NONCE_SALT',       'put your unique phrase here');
```

To generate unique keys and salts using the WordPress.org provided tool [click here](#).

### 4.12.1.4 Further Resources

- [WordPress.org page on keys and salts](#).

## 4.13 Add SSL To The Admin Area

### 4.13.1.1 What You Need To Do

If your host provides a shared SSL certificate or if you have your own SSL certificate you should enable SSL for the WordPress administration panel.

### 4.13.1.2 Why This Point Is Important

If you can connect to your WordPress administration panel using SSL all communication to and from your WordPress site will be encrypted making it much more secure. If you are connecting to your WordPress site from a public wireless network without SSL your login information will be sent in clear text. It is relatively easy for someone to listen in on your communication thus picking up your login details. If you use SSL this will be practically impossible.

**Tip!** If you can't use SSL we recommend you use the [Semisecure Login Reimagined](#) plugin as described on page 30 to at least encrypt your login details.

### 4.13.1.3 How You Complete This Security Checkpoint

Contact your host to ask if they provide an SSL certificate.

If they do add this code towards the top of your wp-config.php file:

```
/* BEGIN WordPress Security Checklist Addition: SSL */
define('FORCE_SSL_LOGIN', true);
define('FORCE_SSL_ADMIN', true);
/* END WordPress Security Checklist Addition: SSL */
```

### 4.13.1.4 Recommendation

Whether you want to purchase an SSL certificate or not will come down to a financial decision.

## 4.14 Monitoring

Using plugins like the [WordPress File Monitor Plus](#) is a great way to keep track of what's going on with your site. However the plugin needs to be active on your site to send you email notifications.

If a hacker gains access to your WordPress administration panel he could disable the plugin, and you would not be notified.

We will address this problem by using monitoring tools that run outside of your WordPress site.

We recommend that you use at least two of the monitoring tools described in this section as they give you different data and work in different ways.

### 4.14.1 WebsiteDefender

In addition to the security features provided by the WordPress plugin as discussed in [WebsiteDefender WordPress Security](#) on page 37 they also offer an online scanner.

This scanner checks for malware, change of content, creation of WordPress administrator accounts and many other things. You can see the extensive list of features [here](#).

#### 4.14.1.1 What You Need To Do

Create an account with WebsiteDefender and setup online scanning of your WordPress site.

#### 4.14.1.2 Why This Point Is Important

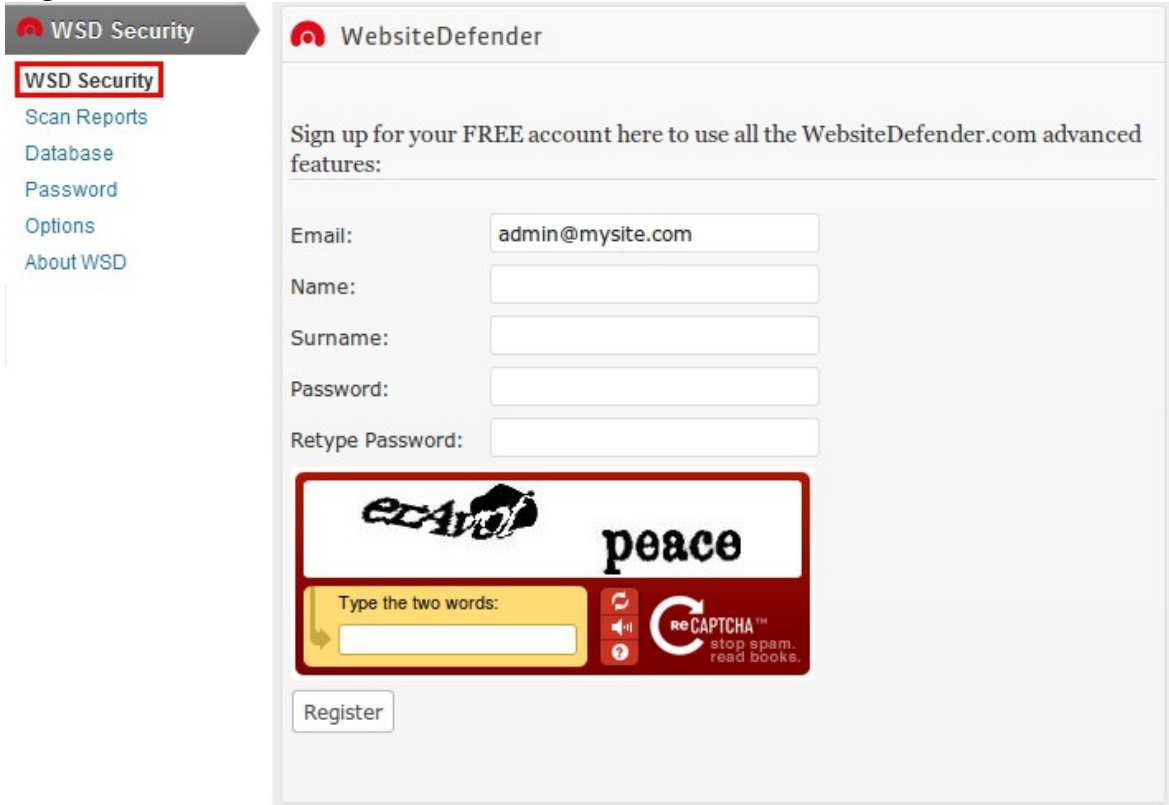
This check runs independently of your WordPress site. Even if your site completely stops working this scan will still run.

#### 4.14.1.3 How You Complete This Security Checkpoint

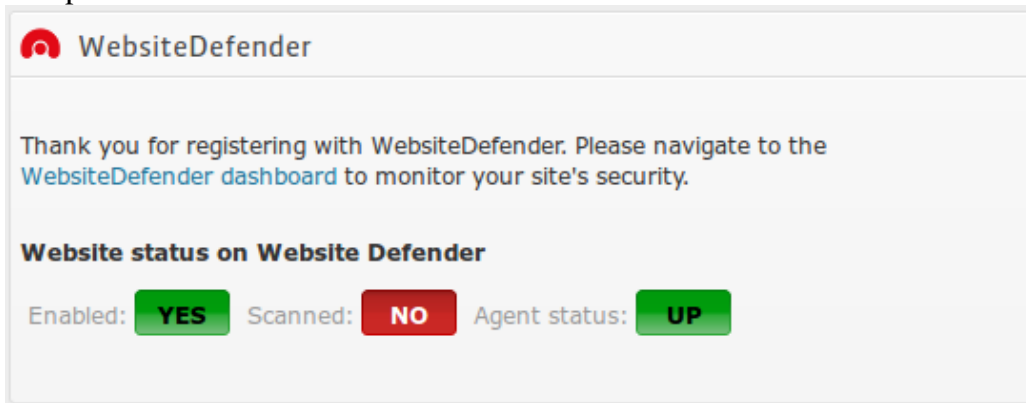
To enable the online scanner follow these steps:

- Install and enable the [WebsiteDefender WordPress Security](#) plugin.

- Register for a free account for the online scanner.



- The status changes.  
Your first WordPress site scan is scheduled, and you will receive an email when it has completed.



- First scan email.



#### YOUR WEBSITEDEFENDER SECURITY REPORT

The latest WebsiteDefender scan on [www.████████.com](#) has detected the following security issue/s:


##### Informational alerts:

- [First scan completed](#)

- On your dashboard at <https://dashboard.websitedefender.com> you can adjust the scan settings.




- Recommended:** Enter a text pattern to search for on your home page. This could be the title of your WordPress site or a key piece of text that will rarely change. If the scan cannot find this text on your home page it will alert you by email. This is a good way to detect if your site has been changed without your knowledge.

 Please specify a text pattern which can be found on your homepage. A text pattern is a few words that appear within the content of your homepage, such as "Welcome to my website". If WebsiteDefender does not find the text pattern you have specified, it means your website is unreachable, either due to your hosting service being down, or another issue. Don't worry, WebsiteDefender will inform you immediately, letting you know what the problem is.

Enter text pattern:

- Optional:** Exclude directories from the scan. This could be the caching directories for example. The default for W3Total Cache is `/wp-content/w3tc/*`.

 If you want to exclude directories or files from the scan, specify them in the below list. To exclude a directory called `/private`, specify `/private/*`. To exclude a file, specify the full path of the file and the filename, e.g. `/blog/test.html`. To exclude files by extension, specify `*.extension`, e.g. to exclude all html files specify `*.html`.

Exclusion path:





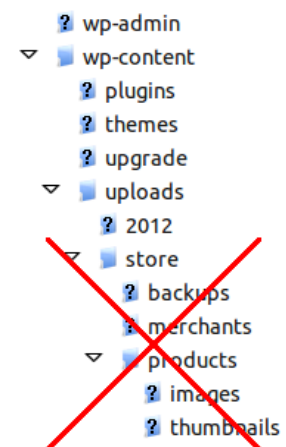
### Datafeedr Tip! ([What is Datafeedr?](#))

If you are using Datafeedr you should be storing your product image files locally (for performance reasons).

If you have a large store with many products you might receive many alerts on new product images being cached locally.

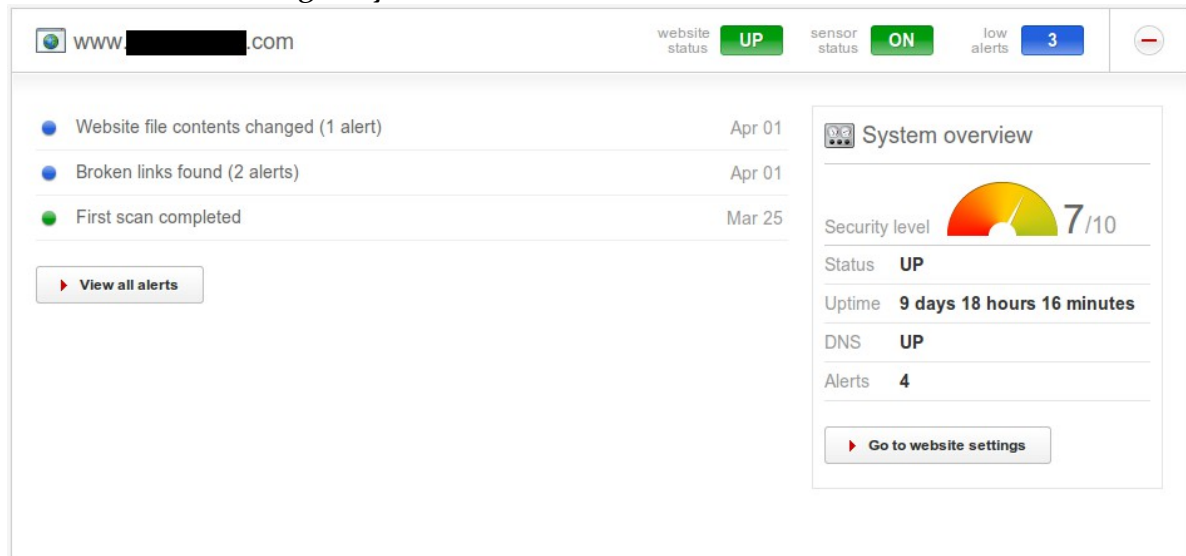
You may choose to exclude the store directory from the scan by adding `/wp-content/uploads/store/*`.

We recommend that you include the store directory.





- The main dashboard gives you a nice overview.



## 4.14.2 Pingdom

Web site: <http://pingdom.com/>

### 4.14.2.1 What You Need To Do

Create an account with Pingdom and setup monitoring for your WordPress site.

### 4.14.2.2 Why This Point Is Important

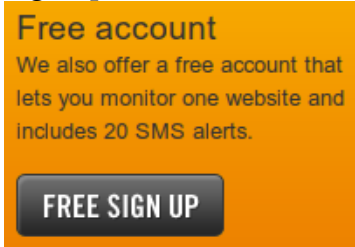
Pingdom is another online monitoring service, which will check for the presence of a key text on your home page.

Additionally they check your site is available with 5 minute intervals, and provide a nice monthly report with your average response times.

### 4.14.2.3 How You Complete This Security Checkpoint

Follow these steps:

- Go to <http://pingdom.com/>
- Sign up for a free account.



- On the **Optional settings** tab enter a string to check for.

The screenshot shows the "Optional settings" tab selected in the Pingdom setup interface. The tab is highlighted with a red box. Below the tab, the section "2. Check settings" contains several input fields. The "Check for string on page:" field is highlighted with a red box and contains the text "Should contain" with a dropdown arrow. Below this field is an empty text input box.

- Remember to **Test** your check before you save.



🟢 Check OK from Montreal, Canada. Response time 0.387 s.

- If this string is not present or if your site is unavailable you will receive an email (or optionally an SMS).

### 4.14.3 Change Detection

Web site: <http://www.changedetection.com/>

#### 4.14.3.1 What You Need To Do

Create an account and setup monitoring for your site.

#### 4.14.3.2 Why This Point Is Important

Change Detection will check for changes to a page on your WordPress site – as opposed to checking for the presence of a key text.

#### 4.14.3.3 How You Complete This Security Checkpoint

Follow these steps:

- Go to <https://www.changedetection.com/>
- Create an account.



- Enter the page you wish to monitor and click **Next**.

**monitor a page**

Page Address:

- Note the options.  
You might want to try out different settings here until you are happy.  
The “sizable change” option tends to work well, as it ignores smaller changes such as dates etc.

☐ only send alert if sizable change

☐ only send alert if text: ☒ added ☐ removed

only send alert if ☐ added ☐ removed text contains these words:

- You will now receive emails with notifications whenever the page changes.

- You can study the details of the changes online, so you can see exactly what has changed.

Current status: normal - monitoring

Number of users monitoring this page: 1

Last notified: 2012-04-02 15:01

Most recent sizable change ([text](#), [html](#)) : 2012-04-02 15:01

Most recent change ([text](#), [html](#)) : 2012-04-02 15:01

Last checked for changes: 2012-04-03 19:13

You started monitoring: 2012-03-15 13:51

Adult content: no

Spam page: no

## 4.15 Cloudflare For Security

Web site: <https://www.cloudflare.com/>

### 4.15.1.1 What You Need To Do

Add a first line of defense to your WordPress site.

### 4.15.1.2 Why This Point Is Important

Cloudflare has advanced security features. They screen all incoming traffic to your WordPress site before it reaches your site. Known threats are automatically blocked.

Cloudflare can help protect you from comment spam, excessive bot crawling, malicious attacks like SQL injection and denial of service (DOS) attacks and more.

For more details visit [this page](#).

In addition to the security features Cloudflare also acts as a Content Delivery Network taking a lot of the load off your site by caching static content.

### 4.15.1.3 How You Complete This Security Checkpoint

Although these steps can seem very technical, they are fairly straightforward and should only take about 5 to 10 minutes to complete.

You will have to look at your current DNS Zone settings and update your name servers.

If you are not sure how to do this ask your hosting provider.

Follow these steps:

- Go to <https://www.cloudflare.com/>
- Create an account and follow their setup instructions.  
It should take around 5 minutes to complete.

- On your WordPress blog Add and Activate the Cloudflare plugin.  
Plugin Page: <http://wordpress.org/extend/plugins/cloudflare/>  
Enter the email address and API key from Cloudflare.

Cloudflare Websites Dashboards **Account** Help Log out

Account > My account My account Communication Billing information

## My account

Your current email address is: [REDACTED] [Change](#)

Your current user name is: [REDACTED] [Change](#)

Change your password [Change](#)

Your API key is: [REDACTED] [Regenerate](#)

---

**Plugins**

- Installed Plugins
- Add New
- Editor
- Akismet
- Configuration
- Cloudflare**

**CloudFlare API Key** [\(Get this?\)](#)

**CloudFlare API Email** [\(Get this?\)](#)

Update options »

- **Optional:** If you use W3 Total Cache enable the Cloudflare integration. Tick **Enable** and enter your account details.

Network Performance & Security powered by CloudFlare

CloudFlare protects and accelerates websites. [Sign up now for free](#) to get started, or if you have an account simply log in to obtain your API key. Contact the CloudFlare [support team](#) with any questions.

CloudFlare: ☒ **Enable**

CloudFlare account email:

API key:  ([find it here](#))

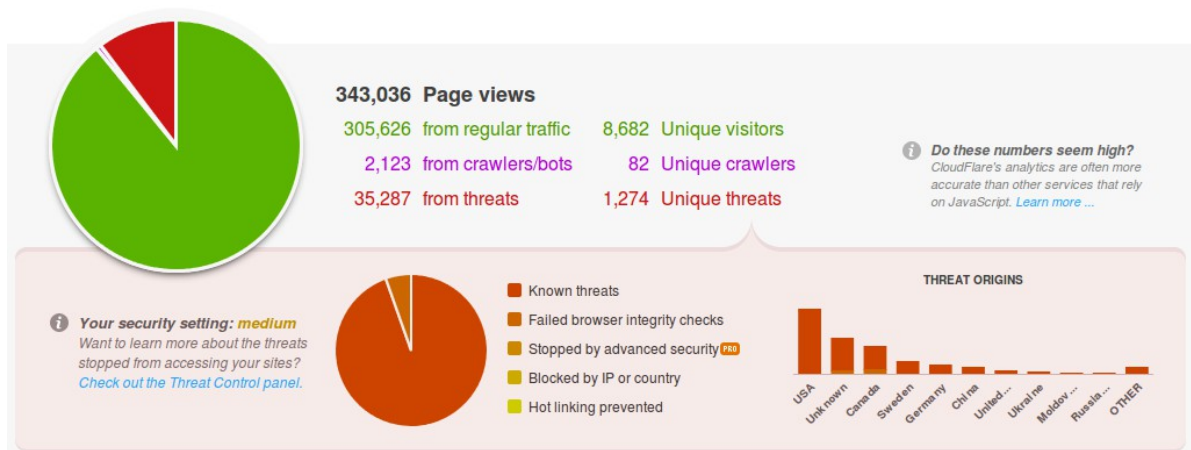
Domain:

Security level: Medium ▾

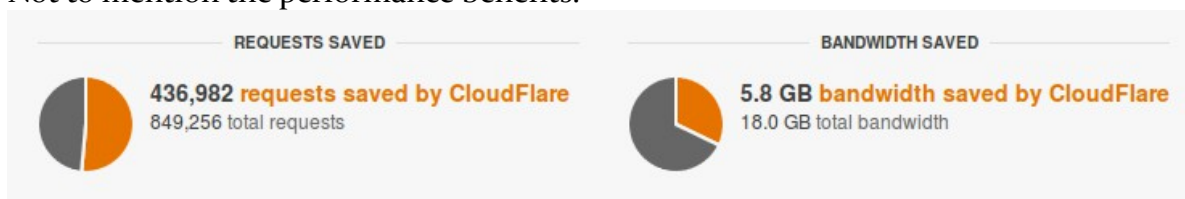
Development mode: Off ▾

[Save all settings](#) [Purge cache](#)

- And enjoy your extra level of security.



- Not to mention the performance benefits.



#### 4.15.1.4 Further Resources

- [Article on how to use Cloudflare.](#)



## 4.16 Enable Logging And Archiving For Apache

### 4.16.1.1 What You Need To Do

Enable logging and archiving of the Apache web server access logs.

### 4.16.1.2 Why This Point Is Important

In case your WordPress site does get compromised the Apache access logs are crucial in determining how the intrusion happened. The log files might help you determine how the intruder entered the site, and thereby pinpoint for example a weak plugin.

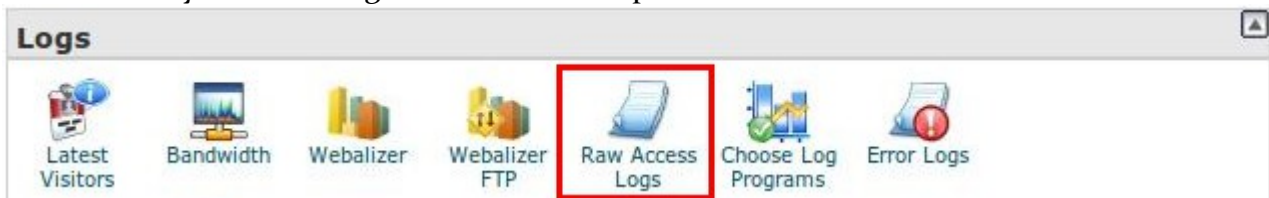
### 4.16.1.3 How You Complete This Security Checkpoint

How you do this depends on your hosting provider. Open a support ticket with them and ask:

*How do I enable Apache access logs and archiving of the logs for my website?*

*For how long are the logs stored?*

On Bluehost you can configure this from the cpanel.



Bluehost log configuration.

#### Raw Access Log

*Raw Access Logs allow you to see who has visited your website without displaying graphs, charts or other graphics. You can use the Raw Access Logs menu to download a zipped version of the server's access log for your site. This can be very useful when you want to quickly be able to see who is visiting your site.*

#### Configure Logs:

- ☒ Archive logs in your home directory at the end of each stats run [ every 24 hour(s)~ ]
- ☐ Remove the previous month's archived logs from your home directory at the end of each month

For a dedicated server you need to setup log rotation. See [this article](#) for more information.

## 4.17 Disable direct access to your database

### 4.17.1.1 What You Need To Do

Your WordPress site consists of files and a database. Normally there is no need to be able to connect to your database from anywhere else but your hosting account (for phpMyAdmin) and from your WordPress site.

If possible you should disable connectivity to your database from any other computers.

### 4.17.1.2 Why This Point Is Important

By disabling remote access to your database hackers will not be able to go directly to your database and inject malicious code. Your database name, user name and password are all stored in the wp-config.php file, and if a hacker has access to that file he can get access to your database.

By blocking remote access you are effectively closing a door into your WordPress site.

Note that not all hosting providers will be able to disable remote access to your database.

### 4.17.1.3 How You Complete This Security Checkpoint

If you are on a shared host you will need to open a support ticket and ask:

*Is it possible to disable remote connectivity to my shared hosting database? And if so how do I do it? I only need my WordPress site and phpMyAdmin from within my hosting account to be able to access my database.*

If you run WordPress on a dedicated server or a virtual private server you can configure MySQL to only allow connections from localhost or a list of known IP addresses.

Godaddy configuration of a new database.

**MySQL Version:**

☒ 5.0

**Allow Direct Database Access** ⓘ

☒ No ☐ Yes

**Note:** Databases with External access are located on a separate server than databases without External Access.

## 4.18 Intrusion Detection Systems

Intrusion Detection Systems monitor system activities and report malicious or suspicious activities.

### 4.18.1.1 What You Need To Do

You should be using an IDS to monitor your system.

### 4.18.1.2 Why This Point Is Important

Detecting when changes occur is a crucial element in determining when a hack has taken place. If you know when an intrusion happened you will be able to select the best backup in case you need to restore your WordPress site.

### 4.18.1.3 How You Complete This Security Checkpoint

If your WordPress site is on a shared host we recommend you use the [WordPress File Monitor Plus](#) plugin described on page 43.

If you are on a dedicated server or a virtual private server we recommend you investigate these options:

- [OSSEC](#)
- [Open Source Tripwire](#)
- [AIDE \(Advanced Intrusion Detection Environment\)](#)

## 4.19 .htaccess files

Please read all of this section before you complete this step.

.htaccess files are configuration files used by the Apache Web Server.

A Web Server is the program that is responsible for taking a request for a page on your website and returning the correct page to the visitor.

Apache is the most commonly used Web Server in the world, and if you are on a shared hosting account you are most likely using an Apache Web Server. The Web Server from Microsoft is called IIS (Internet Information Services) and does not use .htaccess files.

.htaccess files are extremely powerful.

Modifications to your .htaccess file is an effective way of hacking your WordPress site.

For instance a hacker can re-direct all incoming traffic to your website to another website effectively stealing your traffic.

Some hacks redirect only incoming traffic from search engines. This means that if you open your WordPress site by typing the URL or via a bookmark you will see your site working normally. Only users who found your site via search will be redirected. Very sneaky!

We think it is very important that you are familiar with your .htaccess file. Otherwise it is very difficult to detect a hack in this file.

**We highly recommend that you change the .htaccess file manually.**

The changes are not too complicated. You do not need to understand exactly how the code works. Just get a good feel for what the file looks like and get the job done!

The steps to change the .htaccess files manually are described in detail in [How You Complete This Security Checkpoint – Manual Changes](#) on page 85.

**If you are not comfortable making manual changes to your .htaccess files it is much better if you use the plugin instead of doing nothing.**

The [AskApache Password Protect](#) plugin is described in [How You Complete This Security Checkpoint – Using Plugin](#) on page 95.

**Important!** Even a small typing mistake in the .htaccess file can stop your WordPress site from working.

If anything does go wrong you can restore access to your site by restoring the original .htaccess file.

Copy the original .htaccess file to your computer before you start editing.

If you are using the Ask Apache plugin you also need to backup your original .htaccess file. Just in case!

**Important!** When you have completed work on the .htaccess files please ensure you test your WordPress site comprehensively. Especially your contact form, comment posting and any other user input on your site needs to be tested.



#### 4.19.1.1 What You Need To Do

You need to add security measures to the .htaccess file in your WordPress root folder. And you need to password protect your wp-admin folder using the .htaccess file in that folder. You also need to add protection to the wp-includes folder.

#### 4.19.1.2 Why This Point Is Important

Adding security measures in the .htaccess file is very powerful, because the security screening takes place before WordPress even sees the request. This saves a lot of work for WordPress.

#### 4.19.1.3 How You Complete This Security Checkpoint – Manual Changes

##### 4.19.1.3.1 WordPress root folder

##### 4.19.1.3.1.1 Backup The Current .htaccess File

Copy the .htaccess file from your WordPress root folder to your local computer and give it a name like htaccess\_wordpress\_root\_original\_TODAYSDATE.txt.

##### 4.19.1.3.1.2 Update The .htaccess File

Before you start your .htaccess file should look similar to the file below.

```
# BEGIN Cache Plugin
...
# END Cache Plugin

# BEGIN WordPress
<IfModule mod_rewrite.c>
    RewriteEngine On
    RewriteBase /
    RewriteRule ^index\.php$ - [L]
    RewriteCond %{REQUEST_FILENAME} !-f
    RewriteCond %{REQUEST_FILENAME} !-d
    RewriteRule . /index.php [L]
</IfModule>
# END WordPress
```

Don't make changes to the code already in the file.

We will add a few pieces of code marked in bold – explanation follows.

```
# #####
# BEGIN WordPress Security Checklist Addition
# #####

# disable directory browsing
# Note! Bluehost changes this line automatically
Options All -Indexes

# Stop access to sensitive files
# Protect .htaccess
<Files .htaccess>
    Order Allow,Deny
    Deny from all
    Satisfy all
</Files>

# Protect readme.html
<Files readme.html>
    Order Allow,Deny
    Deny from all
    Satisfy all
</Files>

# Protect wp-config.php
<files wp-config.php>
    Order Allow,Deny
    Deny from all
    Satisfy all
</files>

# Protect php.ini
<files php.ini>
```

```

    Order Allow,Deny
    Deny from all
    Satisfy all
</files>

# Protect error_log
<filesMatch "(error_log)$">
    Order Allow,Deny
    Deny from all
    Satisfy all
</filesMatch>

# Block the include-only files
RewriteEngine On
RewriteBase /
RewriteRule ^wp-admin/includes/ - [F,L]
RewriteRule !^wp-includes/ - [S=3]
RewriteRule ^wp-includes/[^/]+\.(php$ - [F,L]
RewriteRule ^wp-includes/js/tinymce/langs/.+\.php - [F,L]
RewriteRule ^wp-includes/theme-compat/ - [F,L]

# #####
#   END WordPress Security Checklist Addition
# #####

# BEGIN Cache Plugin
...
# END Cache Plugin

# BEGIN WordPress
<IfModule mod_rewrite.c>
    RewriteEngine On
    RewriteBase /
    RewriteRule ^index\.php$ - [L]
    RewriteCond %{REQUEST_FILENAME} !-f
    RewriteCond %{REQUEST_FILENAME} !-d
    RewriteRule . /index.php [L]
</IfModule>
# END WordPress

```

## # disable directory browsing

This directive tells the Apache web server to dis-allow directory browsing.

For instance if you go to `www.mysite.com/wp-content/uploads` you will likely see a result like this:

## Index of /wp-content/uploads

- [Parent Directory](#)
- [2011/](#)
- [2012/](#)

*Apache/2.2.21 (Win32) mod\_ssl/2.2.21 OpenSSL/1.0.0e PHP/5.3.8 mod\_perl/2.0.4 Perl/v5.10.1 Server at localhost Port 80*

We do not want our visitors to get this much information about our site.

Now they will see this:

## Access forbidden!

You don't have permission to access the requested directory. There is either no index document or the directory is read-protected.

If you think this is a server error, please contact the [webmaster](#).

Much better!

## # Stop access to sensitive files

We protect a number of files with sensitive information from access from the Internet. These files might expose information such as WordPress version number, server configuration and database prefixes etc.

- `.htaccess` – contains sensitive information about the server configuration and protection.
- `readme.html` – contains the WordPress version number.
- `wp-config.php` – contains information about your database name, user, password and table prefix.
- `php.ini` – contains sensitive information about the server configuration and protection.
- `error_log` – can contain information about your file paths and database table names.

We don't want to give our visitors that type of information.

This is what we will give them:

## 403 Permission Denied

You do not have permission for this request `/error_log`



## # Block the include-only files

This section blocks access to files that no-one should be accessing directly from the Internet anyway. They should only be called from other files on our WordPress site.

### 4.19.1.3.1.3 Test And Backup

Now is a good time to test that your site is still working as it should... especially the parts with user input...

When you have verified everything works copy the updated .htaccess file to your local computer with a name like htaccess\_wordpress\_root\_extra\_secure\_TODAYSDATE.txt.

### 4.19.1.3.1.4 The Perishable Press 5G Blacklist – [web page](#)

The Perishable Press 5G Blacklist is a block of code you add to the .htaccess file. It is very effective at blocking unwanted visitors to your website.

This checkpoint is where you are most likely to see your WordPress site break. If this happens follow these steps:

- Remove all of the 5G code.
- Test to see that your site works again.
- Add the first half of the code.
- Test to see if your site works.
  - If it does work add the second half of the code.
  - If it does not work remove half of the added code and test again.
  - Etc.

By following this method you can quickly determine exactly which line is causing problems. You can then leave out that line. And remember there might be more than one line causing problems.

For more details on this method read [this article](#) on our website.

The more code you leave in the better you are protected. However your site needs to work!

Remember, test thoroughly.

Add this code to the beginning of your .htaccess file:

```
# #####
# BEGIN 5G BLACKLIST/FIREWALL
# @ http://perishablepress.com/5g-blacklist/
```

```
# #####  
# 5G:[QUERY STRINGS]  
<ifModule mod_rewrite.c>  
RewriteEngine On  
RewriteBase /  
RewriteCond %{QUERY_STRING} (environ|localhost|mosconfig|scanner) [NC,OR]  
RewriteCond %{QUERY_STRING} (menu|mod|path|tag)\=\.?/? [NC,OR]  
RewriteCond %{QUERY_STRING} boot\.\ini [NC,OR]  
RewriteCond %{QUERY_STRING} echo.*kae [NC,OR]  
RewriteCond %{QUERY_STRING} etc/passwd [NC,OR]  
RewriteCond %{QUERY_STRING} \=\%\%27$ [NC,OR]  
RewriteCond %{QUERY_STRING} \=\%%' $ [NC,OR]  
RewriteCond %{QUERY_STRING} \%\' $ [NC,OR]  
RewriteCond %{QUERY_STRING} \.? [NC,OR]  
RewriteCond %{QUERY_STRING} \| : [NC,OR]  
RewriteCond %{QUERY_STRING} \[ [NC,OR]  
RewriteCond %{QUERY_STRING} \\ [NC]  
RewriteRule .* - [F]  
</ifModule>  
  
# 5G:[USER AGENTS]  
<IfModule mod_setenvif.c>  
# Next line has been commented out to allow for Google +1 to show images  
# SetEnvIfNoCase User-Agent ^$ keep_out  
SetEnvIfNoCase User-Agent (casper|cmsworldmap|diavol|dotbot) keep_out  
SetEnvIfNoCase User-Agent (flicky|ia_archiver|jakarta|kmccrew) keep_out  
SetEnvIfNoCase User-Agent (libwww|planetwork|pycurl|skygrid) keep_out  
SetEnvIfNoCase User-Agent (purebot|comodo|feedfinder|turnit) keep_out  
SetEnvIfNoCase User-Agent (zmeu|nutch|vikspider|binlar|sucker) keep_out  
<limit GET POST PUT>  
Order Allow,Deny  
Allow from all  
Deny from env=keep_out  
</limit>  
</ifModule>  
  
# 5G:[REQUEST STRINGS]  
<IfModule mod_alias.c>  
RedirectMatch 403 (https?|ftp|php)\:///  
RedirectMatch 403 /(cgi|https?|ima|ucp)/  
RedirectMatch 403 /(Permanent|Better)$  
RedirectMatch 403 (\=\%'|\=\%\%27|/\%'/?|\\)\.css\$()  
# Next line has been modified. Second test ('//') has been removed  
# to allow performance testing using gtmetrix.com and Pingdom FPT  
# RedirectMatch 403 (\,|//|\\)|+|/,/(/{0})|(\/(\\.\\.\\.|\\\\+\\\\+|\\\\\\\\\\\\\\\\\\"\\\\\\\\\\"))  
RedirectMatch 403 (\,|\\)|+|/,/(/{0})|(\/(\\\.\.\.|\\++|\\\\\\\\\\\\\\\\\\"\\\\\\\\\\"))  
RedirectMatch 403 \.(cgi|asp|aspx|cfg|dll|exe|jsp|mdb|sql|ini|rar)$  
RedirectMatch 403 /(contac|fpw|install|pingserver|register)\.php$  
RedirectMatch 403 (base64|crossdomain|localhost|wwroot|el07\_)  
RedirectMatch 403 (eval\\(|\\_vti_|\\(null\\)|echo.*kae|config.xml)  
RedirectMatch 403 \.well-known/host-meta  
RedirectMatch 403 /function\\.array\\-rand  
RedirectMatch 403 \\);\\$(this\\)\\.html\\(  

```

```

RedirectMatch 403 proc/self/environ
RedirectMatch 403 msnbot\.htm\)\.\._
RedirectMatch 403 /ref\.outcontrol
RedirectMatch 403 com\_cropimage
RedirectMatch 403 indonesia\.htm
RedirectMatch 403 \{\$itemURL\}
RedirectMatch 403 function\(\)
RedirectMatch 403 labels\.rdf
RedirectMatch 403 /playing.php
RedirectMatch 403 muieblackcat
</ifModule>

# 5G:[BAD IPS]
<limit GET POST PUT>
    Order Allow,Deny
    Allow from all
    # uncomment/edit/repeat next line to block IPs
    # Deny from 123.456.789
</limit>

# #####
# END 5G BLACKLIST/FIREWALL
# @ http://perishablepress.com/5g-blacklist/
# #####

```

#### 4.19.1.3.1.5 Test And Backup Again

Now is a good time to test that your site is still working as it should...

When you have verified everything works copy the updated .htaccess file to your local computer with a name like htaccess\_wordpress\_root\_extra\_secure\_5G\_TODAYSDATE.txt.

#### 4.19.1.3.2 WordPress wp-admin folder

We will now be working with the .htaccess file in the **wp-admin** folder.

The goal here is to limit access to the WordPress administration panel.

There are two ways of doing this:

1. Limit access to approved ip addresses only.  
 Plus: Very secure. Only computers on specific ip addresses are allowed.  
 Minus: Does not work well if you work from changing locations or if you do not have a fixed ip address.
2. Password protect the wp-admin folder.  
 Plus: Flexible with added protection. Can work from anywhere.  
 Minus: Not as secure. A password can potentially be broken.

You only need to select one method.

If you only ever work with your WordPress administration panel from a few locations with fixed ip addresses we recommend option 1. Otherwise option 2 will work just fine.

Although not quite as secure as option 1 the fact is that most hacking is done by automatic robots working their way through the Internet. They are not particularly intelligent, and if they get turned away from the wp-admin folder they will simply move on to an easier target.

#### 4.19.1.3.2.1 Limit access to approved ip addresses only

**Important!** We are now working on the .htaccess file in the **wp-admin** folder!

If the file does not already exist create it.

If it does exist copy the file to your local computer and give it a name like htaccess\_wpadmin\_original\_TODAYSDATE.txt

Add the following code:

```
# #####
# BEGIN WordPress Security Checklist Addition
# #####

AuthUserFile /dev/null
AuthGroupFile /dev/null
AuthName "WordPress Admin Access Control"
AuthType Basic
<LIMIT GET>
    order deny,allow
    deny from all
    allow from xxx.xx.xx.xxx
</LIMIT>

# #####
# END WordPress Security Checklist Addition
# #####
```

Add your IP address in the **allow from** line. You can add as many **allow from** lines as you wish.

**Tip!** To find your current IP address [click here](#).

Your public IP address is XXXXXXXXXX - [Learn more](#)

**Tip!** If you do need to access your WordPress administration panel from an unlisted ip address you can rename the .htaccess file to .htaccess.bak while you need access... then change the name back when you are done.

**Note!** As mentioned above this restriction only works well if you work from a limited number of fixed ip addresses. Otherwise we recommend you use the password protection described below.

#### 4.19.1.3.2.2 Password protect the wp-admin folder

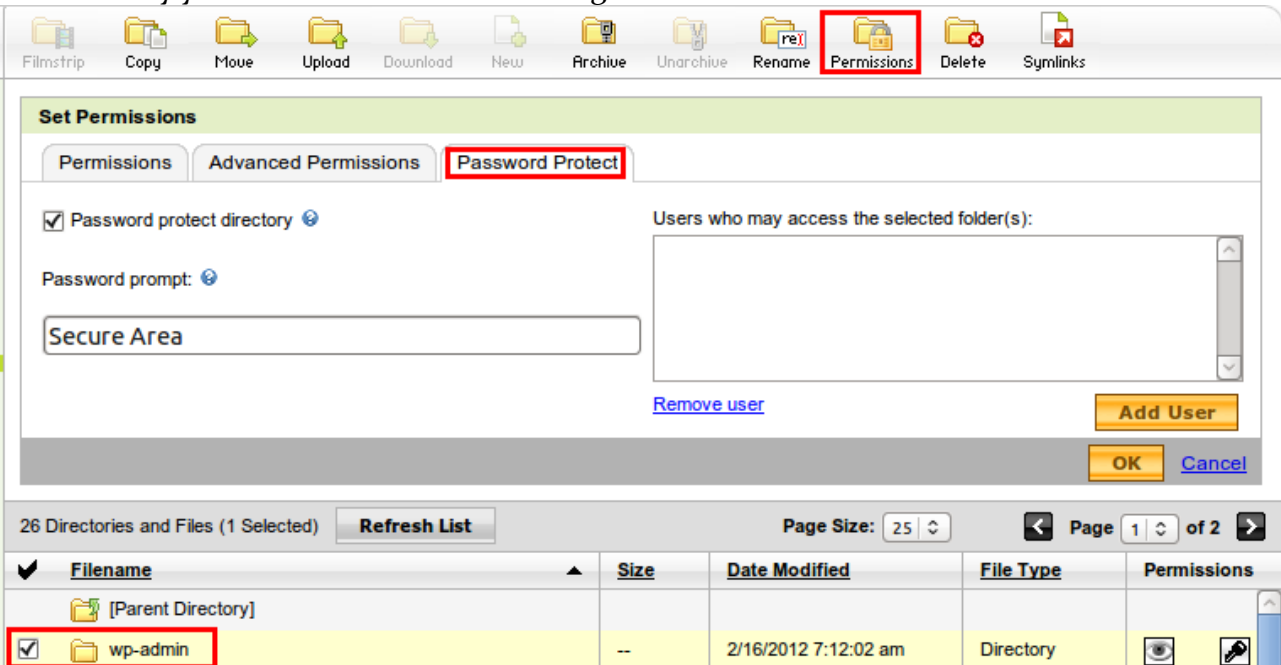
If your WordPress site is on a shared hosting plan you can most likely password protect the wp-admin folder from the cpanel or the File Manager. If you are in doubt ask your host.

This will create the .htaccess file for you and store the user name and password safely.

On Bluehost you select **Password Protect Directories** in the cpanel.

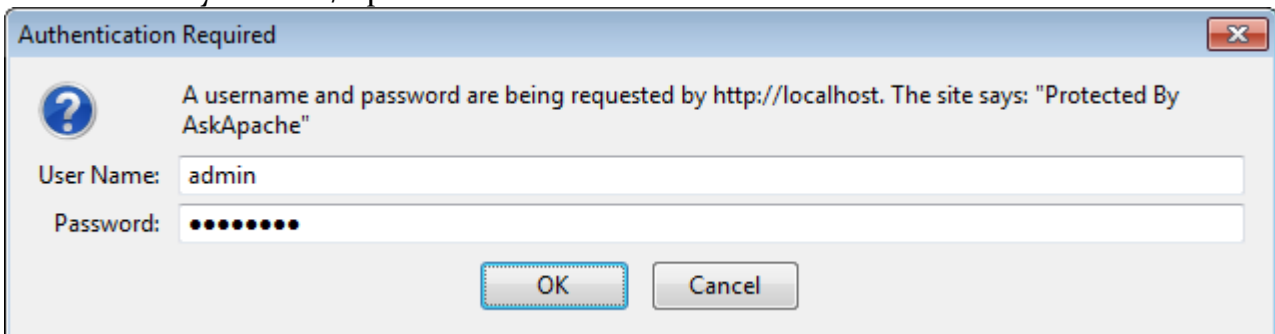


On Godaddy you need to use the File Manager.



If you are on a dedicated server you can find instructions for password protecting a folder [here](#).

Once the wp-admin folder has been password protected you will get this prompt when you access [www.mysite.com/wp-admin](http://www.mysite.com/wp-admin).



We need to allow access to a few files in the wp-admin folder without requiring a password. Open the file and add the lines in bold.

```
AuthType Basic
AuthName "Restricted Access"
AuthUserFile "/xxxxxxxx/passwd"
require valid-user

# #####
# BEGIN WordPress Security Checklist Addition
# #####

<FilesMatch "\.(ico|pdf|flv|jpg|jpeg|mp3|mpg|mp4|mov|wav|wmv|png|gif|swf|
css|js)$">
    Allow from All
</FilesMatch>

<FilesMatch "(async-upload|admin-ajax)\.php$">
    Order allow,deny
    Allow from all
    Satisfy any
</FilesMatch>

# #####
# END WordPress Security Checklist Addition
# #####
```

Save the file and test that you get the password challenge when you access the WordPress administration panel.

## 4.19.1.4 How You Complete This Security Checkpoint – Using Plugin

### 4.19.1.4.1 Backup The Current .htaccess File(s)

Copy the .htaccess file from your WordPress root folder to your local computer and give it a name like htaccess\_wordpress\_root\_original\_TODAYSDATE.txt.

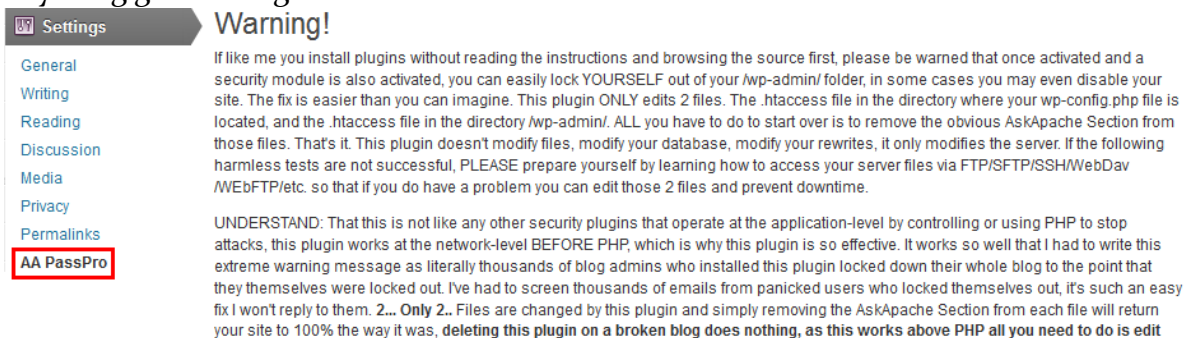
If you already have an .htaccess file in the wp-admin folder copy it to your local computer and give it a name like htaccess\_wpadmin\_original\_TODAYSDATE.txt

**Important!** If you experience any problems using the Ask Apache plugin you can restore access to your WordPress site by restoring these two files... if you did not have an .htaccess file in the wp-admin folder you need to delete the file created by the plugin in case of problems.

### 4.19.1.4.2 Install And Configure The AskApache Password Protect Plugin

Follow these steps:

- Add and Activate the AskApache Password Protect plugin.
- Read the warning, and understand the procedure for restoring your site in case anything goes wrong.



**Warning!**

If like me you install plugins without reading the instructions and browsing the source first, please be warned that once activated and a security module is also activated, you can easily lock YOURSELF out of your /wp-admin/ folder, in some cases you may even disable your site. The fix is easier than you can imagine. This plugin ONLY edits 2 files. The .htaccess file in the directory where your wp-config.php file is located, and the .htaccess file in the directory /wp-admin/. ALL you have to do to start over is to remove the obvious AskApache Section from those files. That's it. This plugin doesn't modify files, modify your database, modify your rewrites, it only modifies the server. If the following harmless tests are not successful, PLEASE prepare yourself by learning how to access your server files via FTP/SFTP/SSH/WebDav /WebFTP/etc. so that if you do have a problem you can edit those 2 files and prevent downtime.

UNDERSTAND: That this is not like any other security plugins that operate at the application-level by controlling or using PHP to stop attacks, this plugin works at the network-level BEFORE PHP, which is why this plugin is so effective. It works so well that I had to write this extreme warning message as literally thousands of blog admins who installed this plugin locked down their whole blog to the point that they themselves were locked out. I've had to screen thousands of emails from panicked users who locked themselves out, it's such an easy fix I won't reply to them. 2... Only 2.. Files are changed by this plugin and simply removing the AskApache Section from each file will return your site to 100% the way it was, deleting this plugin on a broken blog does nothing, as this works above PHP all you need to do is edit

- Scroll to the bottom and click **Initiate Tests**.

Initiate Tests »

- On the Test Results page investigate any items that are not green to see if they are required or not. If you find any problems prohibiting you from running the plugin you can always make the changes manually as described in [How You Complete This Security Checkpoint – Manual Changes](#) on page 85.
- Scroll to the bottom and click **Continue to Setup**.

Continue to Setup »

- Enter the user information to protect the wp-admin folder.

## Setup Password Protection

### Create User

|  |  |
|--|--|
| Admin Email<br>Username and Password sent here<br>in case you forget it. | <input type="text" value="admin@mysite.com"/>  |
| Username   | <input type="text" value="admin"/>   |
| Password (twice)   | <input type="password" value="••••••••"/><br><input type="password" value="••••••••"/> |

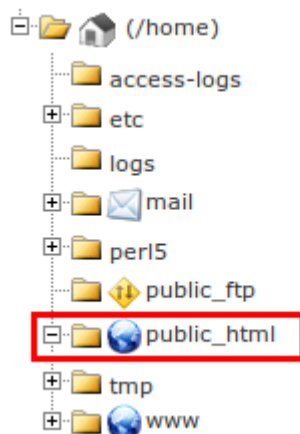
- For the Authentication Settings make sure you select a location for the password file which is **not** under your public\_html folder.

### Authentication Settings

|                        |  |
|------------------------|--|
| Password File Location | <input type="text" value="C:\\xampp\\htdocs\\.htpasswd3"/> |
|------------------------|--|

Use a location inaccessible from a web-browser if possible. Do not put it in the directory that it protects.

If required create a folder at the same level as public\_html.



- The default values should be fine for the remaining fields.
- Click **Save Settings**.



- For **Password Protect wp-admin** click **Activate**.

#### Password

| Name  | Description   | Response                   | Apache Modules | File     | Action                   |
|---|---|----------------------------|----------------|----------|--------------------------|
| <a href="#">Password Protect wp-login.php</a> | Requires a valid user/pass to access the login page..   | 401 Authorization Required | core           | root     | <a href="#">Activate</a> |
| <a href="#">Password Protect wp-admin</a>     | Requires a valid user/pass to access any non-static (css, js, images) file in this directory... | 401 Authorization Required | core           | wp-admin | <a href="#">Activate</a> |

- For **Directory Protection** click **Activate**.

#### General

| Name                                 | Description   | Response | Apache Modules | File | Action                   |
|--------------------------------------|---|----------|----------------|------|--------------------------|
| <a href="#">Directory Protection</a> | Enable the DirectoryIndex Protection, preventing directory index listings and defaulting. | none     | core           | root | <a href="#">Activate</a> |
| <a href="#">Loop Stopping Code</a>   | Stops Internal Redirect Loops   | none     | core           | root | <a href="#">Activate</a> |

- We recommend that you **do not activate** the WordPress Exploit items.

#### Wordpress Exploit

| Name                                | Description  | Response      | Apache Modules | File | Action                   |
|-------------------------------------|--|---------------|----------------|------|--------------------------|
| <a href="#">Protect wp-content</a>  | Denies any Direct request for files ending in .php with a 403 Forbidden.. May break plugins/themes | 403 Forbidden | mod_rewrite    | root | <a href="#">Activate</a> |
| <a href="#">Protect wp-includes</a> | Denies any Direct request for files ending in .php with a 403 Forbidden.. May break plugins/themes | 403 Forbidden | mod_rewrite    | root | <a href="#">Activate</a> |
| <a href="#">Common Exploit</a>      | Block common exploit requests with 403 Forbidden. These can help alot, may break some plugins.     | 403 Forbidden | mod_rewrite    | root | <a href="#">Activate</a> |

### 4.19.1.4.3 Test And Backup

Now is a good time to test that your site is still working as it should... especially the parts with user input...

When you have verified everything works copy the updated .htaccess files to your local computer with names like htaccess\_wordpress\_root\_extra\_secure\_TODAYSDATE.txt and htaccess\_wp-admin\_extra\_secure\_TODAYSDATE.txt..

#### 4.19.1.4.4 Optional

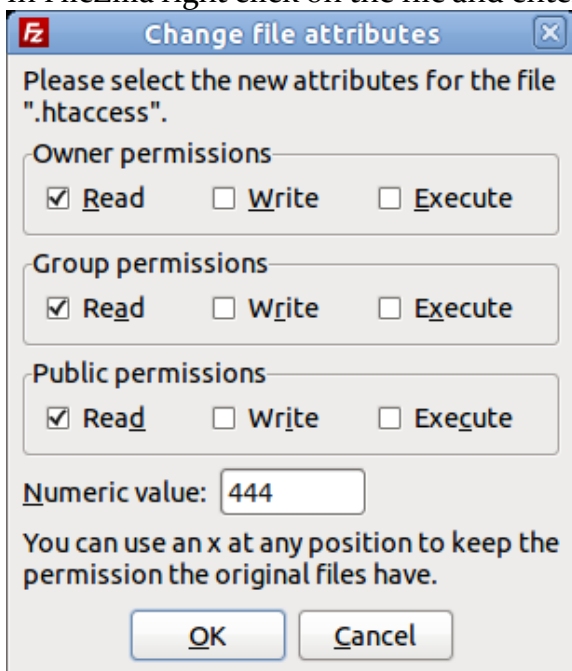
You can activate other modules to help prevent spam and other exploits.

#### 4.19.1.5 Recommendation

When you have made the required changes to the .htaccess files we recommend that you make a backup of the files.

You should also change permissions on all .htaccess files to 444. Changing permissions has successfully stopped some – but not all – attempts at modifying the .htaccess file.

In FileZilla right click on the file and enter 444.



And we **strongly** recommend that you thoroughly test all aspects of your site still work.

#### 4.19.1.6 Further Resources

- [A great article on .htaccess files](#)
- [Article on the 5G blacklist](#)
- [Article on the Perishable Press Halving Method](#)

## 4.20 Securing PHP

PHP is the programming language WordPress is developed in. You can strengthen your WordPress site by disabling certain functions and options in the PHP configuration file. The file is called `php.ini`.

If you are on shared hosting your hosting provider should allow you to have your own `php.ini` file. The configuration can vary from host to host, so ask them if you are in doubt.

If you are using a dedicated server you will be able to make modifications to the `php.ini` file.

### 4.20.1.1 What You Need To Do

Secure your PHP to limit access to your server.

### 4.20.1.2 Why This Point Is Important

Your WordPress site will constantly be probed by robots, that will try to:

- submit malicious data and scripts to your site
- make your server execute malicious scripts located on external servers
- read and write files on your server

They want to take control of your site and use it for their own purposes.

### 4.20.1.3 How You Complete This Security Checkpoint

You need to change a few settings in your `php.ini` file to make PHP more secure.

Note that this might break some plugins if they require some of these functions to work properly. Therefore it is important that you test your site after making these changes to verify everything is working. If you encounter a problem try changing one setting at a time to its original value until you find the setting that causes the problem.

If you need that particular plugin then you have to live with a slightly less secure configuration of PHP. You are still miles ahead of most other web sites.

- Ask your host about your options for modifying the `php.ini` file. Typically there is one `php.ini` file in the `public_html` folder, however some hosts allow you to have multiple `php.ini` files.

- Locate your php.ini file.  
If it does exist find each setting (e.g. search for **short\_open\_tag**) and change the value.  
If it does not exist create a text file, name it php.ini and add the code below.

```
; BEGIN WordPress Security Checklist Additions

register_globals = Off
allow_url_fopen = Off
short_open_tag = Off
display_errors = Off
display_startup_errors = Off
log_errors = On
magic_quotes_gpc = Off
magic_quotes_sybase = Off

; If you experience problems after changing the php.ini the line below
; is the place to look first.
disable_functions = show_source, system, passthru, exec, phpinfo, popen,
proc_open

; END WordPress Security Checklist Additions
```

**Test!** Especially plugins that allow user input, e.g. forums, commenting, contact forms, galleries etc. If you do find any plugins have stopped working you are most likely to find the problem in the `disable_functions` line.

Comment out the whole line by adding a semicolon (;) at the beginning of the line.

If the plugin starts working try taking out each function one at a time until you find the one (or two) that stop the plugin from working. Leave in as much as possible.

#### 4.20.1.4 Explain what the steps do

```
register_globals = Off
```

**Important setting!** Setting `register_globals` to Off will make it harder for someone to inject code as variables in php cannot easily be changed.

```
allow_url_fopen = Off
```

**Important setting!** Setting `allow_url_fopen` to Off will stop WordPress from including and executing code from other websites in the php code. This is a very common way of injecting malicious code into a website.

```
short_open_tag = Off
```

Setting `short_open_tag` to Off will make php code interpretation more strict. Some types of poorly written code will be rejected.

```
display_errors = Off
display_startup_errors = Off
log_errors = On
```

These settings make sure WordPress does not display error information publicly. Error information can be a great source for hackers to discover intimate information about your server configuration.

Instead errors will be logged to a file.

**Note!** If you had an existing `php.ini` file check the setting for `error_log`. This sets the file name for the error log. In case the file name is not `error_log` you will need to modify the `.htaccess` file discussed in [# Stop access to sensitive files](#) on page 88 accordingly.

Example:

```
; Log errors to specified file.
error_log = error_log
```

```
magic_quotes_gpc = Off
magic_quotes_sybase = Off
```

This setting will tell php to apply a strict interpretation on the use of quotes in the code.

```
disable_functions = ...
```

This setting tells php to turn off certain powerful functions, which are typically used by hackers. However some of these functions can also be used legitimately by some plugins.

Therefore you need test your plugins carefully after enabling this.

#### 4.20.1.5 Recommendation

The more of these settings you can successfully apply the better. However you should not sacrifice required functionality on your WordPress site in case there is a problem with one of these settings.

#### 4.20.1.6 Further Resources

[Excellent article on securing PHP.](#)



Want know when we recommend a new plugin?

[Sign up for our newsletter Now!](#)

## 5 Rescue Plan

So what should you do in case the alarm goes off?

Well, now you are well prepared.

With the monitoring tools you have employed you will be notified as soon as an intrusion is detected.

Assuming you have employed the backup plan we recommended in [Schedule Backups Of Your WordPress Site](#) on page 54 you should have a number of good backups to choose from.

Say you run a daily backup at 3am and your File Monitor Scan runs daily as well. Note that in the default setup of the File Monitor you cannot control the time of the scan. The scan will only be triggered when you have visitors to your website, so the time may vary.

Say for example that today is Thursday 21 April.

You get an alert from the File Monitor plugin that some files have been added to your site. The file dates are from Wednesday 20 April and the time of the email is 8pm.

| Day                | Time | Action                 |
|--------------------|------|------------------------|
| Monday 18 April    | 3am: | BackWPup               |
|                    | ??m: | File Monitor Scan      |
| Tuesday 19 April   | ??m: | File Monitor Scan      |
|                    | 3am: | BackWPup               |
| Wednesday 20 April | ??m: | File Monitor Scan      |
|                    | ??m: | Suspicious Files Added |
|                    | 3am: | BackWPup               |
| Thursday 21 April  | 3am: | BackWPup               |
|                    | 8pm: | File Monitor Scan      |

Because we do not have control over the time of the File Monitor Scan the safe choice is to work with the backup file from Monday 18 April. This places one full day between the day of the intrusion and the backup file we determine as being safe.

We do not know the order of events on the Tuesday and Wednesday. The File Monitor Scan might have taken place before the backup Tuesday. The infection could have taken place before the Wednesday backup. The file dates and times on the suspicious files are not a safe indication of when the intrusion actually took place.



Also you need to be aware of differences in time zones between your WordPress host and your own computer.

You could inspect the contents of the Tuesday backup file to see if it includes the suspicious files reported by the File Monitor Scan. If not the backup is most likely safe.

If you know that your site has not been updated with new posts or user registrations we recommend that you go a few more days back.

If your site has been updated Tuesday or Wednesday we recommend that you manually keep track of the changes, e.g. copy the contents of new posts to a file on your computer. When you have restored the backup you need to recreate the posts.

If new users have registered on your site you will need to email them and politely ask them to register again. Or you can recreate the users and email a new password to them.

Before you restore your backup you should manually run a backup of your site as it is now. Save the backup in a safe place and clearly mark it as infected. Then remove the backup file from your hosting account and from your Dropbox. You do not want to confuse this backup with a clean backup at a later point in time. You will need this backup if you wish to investigate how the intrusion happened. You should also move any other backups taken since the safe backup, and note them as being potentially infected. In our example that would be the backups from Tuesday 19 April, Wednesday 20 April and Thursday 21 April.

Now you need to restore the safe backup from before the intrusion happened. You can find instructions for site restoration in [WordPress Site Restoration](#) on page 12.

Please be aware that the WebsiteDefender scan only runs once per week unless you have a paid account.

If WebsiteDefender reports an intrusion you will have to go back at least one week to find a safe backup. In the example above you'd have to go back to the backup from Wednesday 13 April.

**Important!** You need to do a complete site restoration including your database because you do not know if the intrusion also modified the contents of your database.