WordPress Security Tips

Peter edited this page on Jan 6, 2015 · 31 revisions

These WordPress security tips are listed in no particular order. This is a working document on a wiki - feel free to contribute.

For further WordPress hardening tips see: http://codex.wordpress.org/Hardening_WordPress

1. Keep the blog on a subdomain.

This will ensure your blog is on a different 'origin' than your main website. The browser's Same Origin Policy (SOP) will add some client side separation between the two. This may, however, effect your blog's Google page rank.

2. Move the wp-content directory.

Moving the wp-content directory will help protect your blog against some automated attacks.

> Since Version 2.6, you can move the wp-content directory, which holds your themes, plugins, and uploads, outside of the WordPress application directory.
> http://codex.wordpress.org/Editing_wp-config.php#Moving_wp-content

3. Do not use the 'admin' username.

WordPress used to set the 'admin' username by default on all installations. In recent versions the username can now be chosen on installation. Since it is widely known that a lot of WordPress blogs use the 'admin' username it is a prime target for password brute force attacks.

4. Keep plugin installations to a minimum.

Through experience we have found that WordPress plugins are normally the weakest link in a WordPress blog's security. Many plugins are susceptible to Cross-Site Scripting (XSS), SQL Injection and other attacks. By keeping plugin installations to a minimum you reduce the attack surface.

5. Move the wp-config.php file one directory up, outside of the web root directory.

WordPress will look inside the web root directory for the wp-config.php file as well as within the directory above it. This will help in minimising the file being exposed to the Internet.

6. Turn off verbose errors on your web server.

WordPress suffers from many Full Path Disclosure (FPD) vulnerabilities which can be used to facilitate in further attacks such as Path Traversal. A bandaid for these bugs is to turn off

WordPress CVEs

WordPress Security Tips

WordPress Versions Release Dates

WPScan Documentation

Clone this wiki locally

verbose errors in your web server's configuration file.

Solution: disable PHP reporting. Add this line in the 'php.ini' file.

```
error_reporting  =  off
```

7. Remove any TimThumb files.

TimThumb is a small php script for cropping, zooming and resizing web images which many WordPress themes use. In 2011 a Remote Code Execution vulnerability was found to affect it and was actively exploited. The lead developer has since dropped the project.

8. Use a login lockdown plugin.

WordPress by default does not limit the number of unsuccessful login attempts which makes it susceptible to a password brute force attack. There are many plugins which introduce this functionality as well as other login security features.

9. Remove the readme.html file.

Every time WordPress is installed or updated a file called readme.html is included. This file may disclose your blog's version number which could aid an attacker in exploitation.

10. Keep WordPress and its plugins updated.

WordPress and plugin authors are constantly fixing bugs and security issues within their code and releasing new versions. At the time of writing only 21.5% of WordPress blogs are running the latest version.

11. Administration over SSL.

The wp-login.php file is often accessed over un-encrypted channels such as HTTP. By ensuring the connection is encrypted when you submit your login credentials you reduce the risk of Man In The Middle (MITM) attacks. For further information see: http://codex.wordpress.org/Administration_Over_SSL

12. Use unprivileged database user for non-admin functionality. (needs some WP modification)

By default WordPress uses the same database user for all users, anonymous users through to authenticated admins. With some code tweaks it is possible to use a lower privileged database user for anonymous users, reducing the risk of database compromise.

13. Don't use the default 'wp_' table prefix.

By default WordPress uses the 'wp_' database table prefix. This prefix makes it easy for attackers to guess table names. It is recommended that alternative prefixes be used.

14. Keep privileged users to a minimum.

The more privileged users there are the more chance that one of them has a weak password. By keeping privileged users to a minimum you reduce the risk of them being compromised.

15. Remove the version number from the generator tags (index page and RSS feed).

WordPress by default advertises its version in HTML generator meta tags both on the index page and within its RSS feed. By removing these it makes it more difficult for an attacker to find out the blog's version.

16. Add a layer of protection to the wp-admin directory and the wp-login.php file with HTTP Basic Authentication.

17. Disable the theme and plugin editor.

Attackers whom have gained access to the theme/plugin editor could use them to execute their own malicious code. The theme and plugin editor can be disabled within the wp-config.php file.

18. Only use plugins from the official directory.

It is recommended that only plugins published on the official WordPress plugins website be installed. WordPress does some plugin vetting which should catch rouge plugins.

19. Don't store backups on public directories.

Backup files should not be stored within a public web directory. The backup file names may be guessed via the use of brute force techniques.

Solution: switch off debugging. Change these lines in the 'wp-config.php' file:

```
// Switch off debugging.
define('WP_DEBUG', false);
// Switch off debug logging.
define('WP_DEBUG_LOG', false);
```

20. Remove 'Powered by WordPress' from the blog's footer.

Attackers may use search engines to find potential vulnerable victims. By removing the 'powered by' text it may help in preventing basic attacker enumeration.

21. Enable X-Frame-Options for unauthenticated users.

WordPress in recent versions uses the 'X-Frame-Options' HTTP header for privileged users to tell the browser where HTML frames are allowed to be loaded from. This isn't however set for unauthenticated users, allowing for potential ClickJacking attacks.

22. IP whitelist the wp-login.php file.

Most administrative users login to their blog via the same IP address. By whitelisting access to the wp-login.php file you ensure that only specific IPs can access it.

23. Use a strong password

24. Review the Headers sent by your Webserver

You should always review the HTTP Headers sent by your Webserver and limit them to a minimum. To check your Headers you can execute the following command and check the output.

```
curl -skI http://www.domain.com
```

As an example, PHP sends it's version information in a header. To disable this, add or uncomment the following line in your php.ini:

```
expose_php = Off
```

If you are running apache, you can also minimize the info sent about your Webserver. You should edit the file `/etc/apache2/conf.d/security` and set the following values:

```
ServerTokens Prod # Only show Server: Apache
ServerSignature Off # Remove internal information
```

```
TraceEnable Off # Disable trace method
```