How to create a secure WordPress install v1.1

By BlogSecurity.net

Table of Contents:

Introduction

This paper provides you with all necessary information to improve the security for your blog. We try to describe the steps in an easy, understandable way without to much tech talk, so that you can easily follow them and you don't run into problems with applying these changes to secure your blog. All information can be found on BlogSecurity.net; this paper serves as a compact guide to secure your blog. We will strive to keep this document updated, so check back regularly.

If you have questions, problems, ideas or something else related to this paper, feel free to contact us.

Important: Before using any of the techniques outlined is this paper please perform a full backup of your WP files and Database. See our "5 failsafe steps to upgrade WordPress" for help.

Installing WordPress

Accessing your WordPress tables

Before you even start to install your blog it is important to choose the right type of database user with the right permissions. The idea behind this "vital" step is to use a limited database user to administrate your blog. In the event that your blog is compromised, the attacker will struggle to escalate his/her privileges outside of your blog (i.e. the rest of the web server or hosted account).

Note: Never let a web application access the database with a **root user**. In short, you are looking for trouble.

Your user-account should have **no global** SQL privileges as well as limited access to the database where your WordPress tables reside. It's enough to give your user the following rights in order to perform all daily tasks:

> *For Data: SELECT, INSERT, UPDATE, DELETE*
> *For Structures: CREATE, ALTER, DROP*

If you're only allowed to run one database with your hosting account, but you can create multiple users within this database, create a user whose access is limited to the WordPress tables spoken of. The privileges should be:

> *For Data: SELECT, INSERT, UPDATE, DELETE*
> *For Structures: ALTER*

Why are we doing this? It's for your security, so if an attacker gains access to your blog and is able to run queries, these queries are then limited only to your WordPress installation and not the rest of your databases.

Changing your WordPress Table Prefix

To mitigate SQL injection threats you should change the default WordPress prefix from wp_ to something **more random** like *4i32a_*. Often attackers use public exploits of the Internet. These exploits will likely rely on the fact that most WordPress installations use the table prefix, "wp_". Changing this will make it more difficult for an attacker to successfully run exploits.

To easily recognize which prefix stands for which web application you can keep something within the prefix of the application, such as *wp_4i32a_* or *wp4i32a_*. It's just important to modify it, so an attacker can't easily guess your WordPress prefix.

Before Installation

If you just start to install WordPress this step is quite easy to achieve, all you need to do is to change this line from the **WP-CONFIG.PHP** file:

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

to something different, for our example:

```
$table_prefix = '4i32a_';
```

After that you can simply run the installation routine of WordPress and all tables and fields are created with your **different** WordPress prefix.

Manually Change

You want to change the table prefix on an existing installation it requires a few additional steps. This use to be a tedious process and explained below, however, for the quick solution see the next section.

The first step is to open your **WP-CONFIG.PHP** file and change that line:

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

To use the previous example again we're using the prefix *4i32a_* so our line should look like this:

```
$table_prefix = '4i32a_';
```

Now that this is done we need to rename all WordPress tables to match the new prefix, to achieve that we need to run a SQL Command[1] within a web interface like PHPMyAdmin or similar, as WordPress doesn't allow it to change the prefix directly.

So these tables:

> *wp_categories, wp_comments, wp_link2cat, wp_links, wp_options, wp_post2cat,*
> *wp_postmeta, wp_posts, wp_usermeta, wp_users*

Should become this:

[1] Example Query: *RENAME TABLE wp_categories TO 4i32a_categories*

> *4i32a_categories, 4i32a_comments, 4i32a_link2cat, 4i32a_links, 4i32a_options, 4i32a_post2cat, 4i32a_postmeta, 4i32a_posts, 4i32a_usermeta, 4i32a_users*

Now you may think that we're done right? But that's not the case. WordPress includes some values with the table prefix as well. In order to be able to use your blog we need to change them first.

Within the table wp_options[2] we need to change the value of one record for the field option_name from *wp_user_roles* to *4i32a_user_roles*[3].

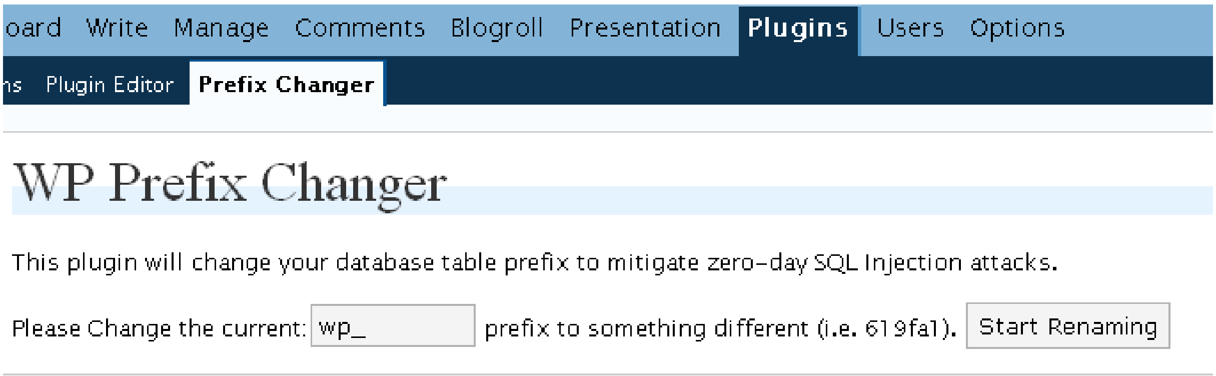
Now you need to replace two[4] other values in this table: wp_usermeta.
The values *wp_autosave_draft_ids* and *wp_user_level* for the field meta_key need to be changed to the new prefix: *4i32a_autosave_draft_ids* and *4i32a_user_level*.

That's it! **But BlogSecurity has made this even easier with** WP Prefix Table Changer.

Through WP Prefix Table Changer

We created a plugin called WP Prefix Table Changer, which automates the whole process of renaming your WordPress tables and the mentioned table entries.

After you downloaded the plugin from BlogSecurity.net, you just need to extract the files into your WordPress plugin-folder, which should be **WORDPRESS/WP-CONTENT/PLUGINS**. After this you need to enter the WP-Admin area and enable the WP Prefix Table Changer Plugin. After the activation a new submenu tab appears within the plugin page menu called Prefix Changer, click on it. On that page you see the following:


oard Write Manage Comments Blogroll Presentation Plugins Users Options
ns Plugin Editor Prefix Changer

WP Prefix Changer

This plugin will change your database table prefix to mitigate zero-day SQL Injection attacks.

Please Change the current: wp_ prefix to something different (i.e. 619fa1). Start Renaming

As you can see you this blog is using the default database WP Prefix. Now change that displayed prefix to something **random, meaningless** like our old example *4i32a_*. Once done, just click the 'Start Renaming' button and the plugin starts to rename all table names from wp_ to your selected random string. It should be noted that 3rd Party Component Tables are also changed, as they need that new prefix too.

[2] The default prefix is used, to avoid confusions that could occur if we would use the new prefix

[3] *UPDATE 4i32a_options SET option_name='4i32a_user_roles' WHERE option_name='wp_user_roles' LIMIT 1*

[4] **Note**: it can be that these fields don't exist currently, as they're just created when they're needed, once they get created they get the correct value automatically

The plugin's last step is to change the prefix within the **WP-CONFIG.PHP** file.

You'll get a message if this process has succeeded or failed. If the process is successful the **WP-CONFIG.PHP** file will be made read-only (**644**) for security purposes, if the process failed the file is mostly unwriteable already. So if the process failed all you need is to open the **WP-CONFIG.PHP** file manually and change the table prefix to your new random prefix.

Preparing the Blog

Now your blog is installed and some excellent security fundamentals are in place, well done! Next we are going to make some changes to your WP-users, in order to improve the security.

Changing your Admin Username

You should rename your default administration account from **admin** to something harder to guess, as all currently available WP versions are vulnerable to User Enumeration. Changing the default admin account to another name will help mitigate password brute force attacks.

Note: You must assume the attacker will know your username. Therefore, ensure that you have chosen a strong password.

Login with your Admin account and create a new user, which holds the role Administrator. Set the username to something that is hard to guess or enumerate. Give him a password (something not so easy to guess, with non alphanumeric characters) and so on, you can use the same email as for your old Admin account, as we are going to delete that.

After the account is created you need to switch the account you're using currently. Now you're logged in with your new admin account. It's time to drop your old admin.

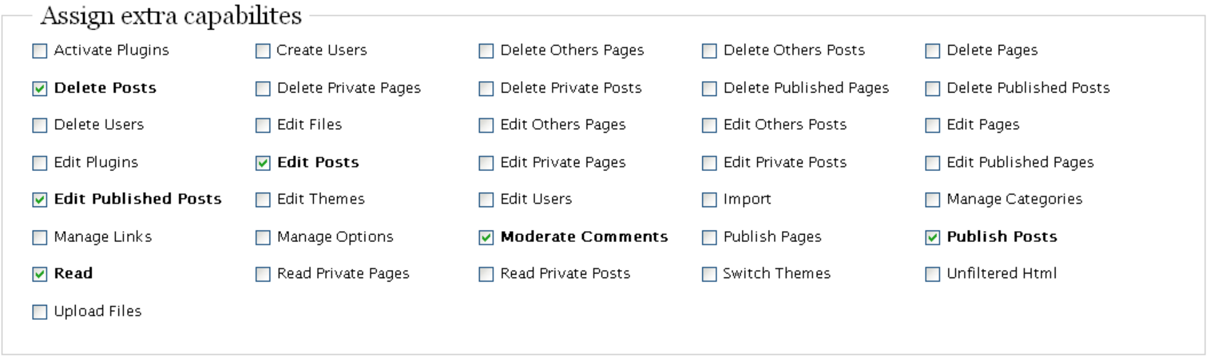
Create a new limited access user

Before we start with that step you should grap yourself a copy of the plugin Role Manager by im-web-gefunden. This plugin will enable you to set granular WP user permissions for each user. After you activated the plugin, create a new user account with the features that you like.

The less that your user account is able to do the better your security will be; your new user should have no higher role than Contributor.

The role of Contributor may not have enough rights by default, therefore we installed the Role Manager plugin before. That plugin allows it to set **additional rights** for each user that he wouldn't normally have with his role, you can change the role rights or even create new roles.

As a standard we suggest new users be given “contributor” level access, however, this plugin allows you a lot more flexibility. As seen below.



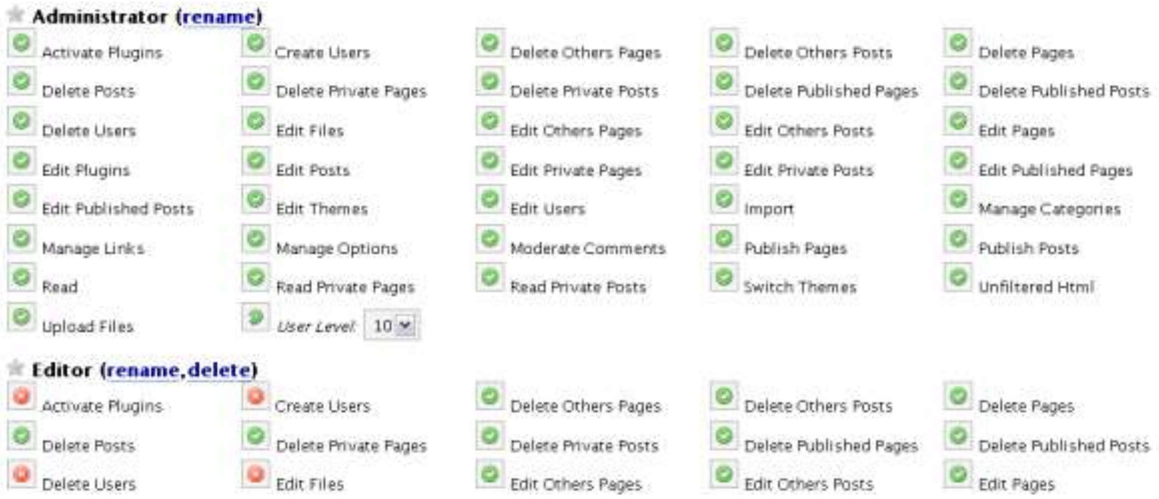
Assign extra capabilites

□ Activate Plugins	□ Create Users	□ Delete Others Pages	□ Delete Others Posts	□ Delete Pages
☑ **Delete Posts**	□ Delete Private Pages	□ Delete Private Posts	□ Delete Published Pages	□ Delete Published Posts
□ Delete Users	□ Edit Files	□ Edit Others Pages	□ Edit Others Posts	□ Edit Pages
□ Edit Plugins	☑ **Edit Posts**	□ Edit Private Pages	□ Edit Private Posts	□ Edit Published Pages
☑ **Edit Published Posts**	□ Edit Themes	□ Edit Users	□ Import	□ Manage Categories
□ Manage Links	□ Manage Options	☑ **Moderate Comments**	□ Publish Pages	☑ **Publish Posts**
☑ **Read**	□ Read Private Pages	□ Read Private Posts	□ Switch Themes	□ Unfiltered Html
□ Upload Files				

See: Role Manager information.

If you have multiple users, it is best to change the rights for the additional role, or maybe you should even create a new role as you want to run some users within normal roles as well.

When creating users, be wary of allowing untrusted users access to roles such as “upload files”, “general plugin access”, “edit files/pages/posts”, “import” and “unfiltered html”, as these roles give the user a lot of power. In two words, be wise.



Administrator (rename)

✓ Activate Plugins	✓ Create Users	✓ Delete Others Pages	✓ Delete Others Posts	✓ Delete Pages
✓ Delete Posts	✓ Delete Private Pages	✓ Delete Private Posts	✓ Delete Published Pages	✓ Delete Published Posts
✓ Delete Users	✓ Edit Files	✓ Edit Others Pages	✓ Edit Others Posts	✓ Edit Pages
✓ Edit Plugins	✓ Edit Posts	✓ Edit Private Pages	✓ Edit Private Posts	✓ Edit Published Pages
✓ Edit Published Posts	✓ Edit Themes	✓ Edit Users	✓ Import	✓ Manage Categories
✓ Manage Links	✓ Manage Options	✓ Moderate Comments	✓ Publish Pages	✓ Publish Posts
✓ Read	✓ Read Private Pages	✓ Read Private Posts	✓ Switch Themes	✓ Unfiltered Html
✓ Upload Files	✓ *User Level* 10			

Editor (rename,delete)

✗ Activate Plugins	✗ Create Users	✓ Delete Others Pages	✓ Delete Others Posts	✓ Delete Pages
✓ Delete Posts	✓ Delete Private Pages	✓ Delete Private Posts	✓ Delete Published Pages	✓ Delete Published Posts
✗ Delete Users	✗ Edit Files	✓ Edit Others Pages	✓ Edit Others Posts	✓ Edit Pages

Changing the role rights with Role Manager

Hardening your WP Install[5]

This will cover how to protect your admin area from unauthorised access. This step is easier to achieve for single user blogs, and can be a real pain for multiple user blogs. You have to decide if you want to shoulder that in order to secure your blog or not, but you should.

Restricting wp-content & wp-includes

In this step we're going to limit the access to these directories, we're mainly going to generally deny everything, except the request for images, CSS and some JavaScript files.

You need to put this code into your **.HTACCESS** file for the folders **WP-CONTENT** & **WP-INCLUDES**:

```
Order Allow,Deny
Deny from all
<Files ~ ".(css|jpe?g|png|gif|js)$">
 Allow from all
</Files>
```

Note: You may want add "specific" PHP file(s) access for certain templates and plugins.

Restricting wp-admin

Block all except your IP

If you run a single user blog, you may want to restrict access to your **WP-ADMIN** directory via IP. Make sure you have a static IP address (one that doesn't change) before doing this. The **.HTACCESS** file within **WP-ADMIN** should look like that:

```
Order deny,allow
Allow from a.b.c.d #That's your static IP
Deny from all
```

Save the file and try to access the wp-admin folder through a web proxy; it should be blocked if everything is working correctly. After that try access it directly with your IP (a.b.c.d).

If everything works correctly **WP-ADMIN** will now be restricted only to the IP(s) of your choice.

Password Required - .htpasswd

Certainly the preferred option is to use password protection. This means you can still access your admin directory from anywhere, however, it adds an additional security layer.

The .htaccess file

he **.HTACCESS** file within **WP-ADMIN** should look like that:

```
AuthUserFile /srv/www/user1/.htpasswd #this file should be outside your webroot.
AuthType Basic
```

[5] The original article can be found here: http://blogsecurity.net/WordPress/article-210607/

```
AuthName “Blog”
require user youruser #making this username difficult to guess can help mitigate password brute force attacks.
```

The .htpasswd file

This file[6] should, as already mentioned, be placed somewhere out of your web directory, ideally one folder above it. To generate the password encrypted you can use: http://www.euronet.nl/~arnow/htpasswd/ but many others are available as well, simply input your username and your plain password into this form and then write the outputted code into the **.HTPASSWD** file, your file should now look like this:

```
Yourusr:$a983seJ/a25.Aa
```

Now test it to see if everything is working; if that isn’t the case, rewrite the encrypted password.

MUSTHAVE Plugins

Not every plugin is potentially a additional security risk. This list of plugins can greatly improve the security of your blog.

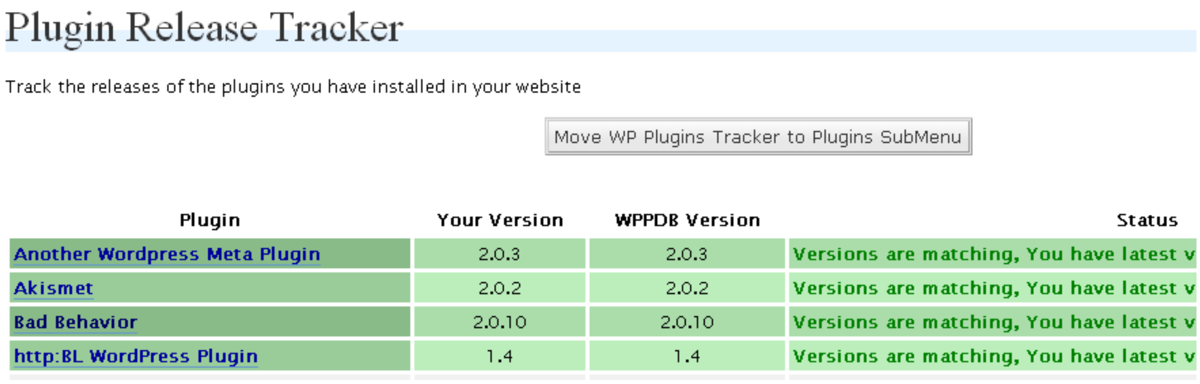
WPIDS - Detect Intrusions

BlogSecurity have ported PHPIDS (Intrusion Detection System) into WordPress. PHPIDS is able to detect various intrusion attempts. We use this facility to block dangerous attacks. Every intrusion is logged in the database, so you can keep track and take the necessary steps if needed. You’re able to get an email if the impact was bigger than a defined value (every intrusion has its own threat score). You can also block the attacker’s IP for a given amount of days if the threat isn’t scored lowly, however, in all cases WPIDS will attempt to sanitise bad input. You can get your copy from the official PHPIDS website. Note: In order to be able to use this plugin you need to run at least PHP 5.1.6 or higher. A new version of WPIDS will be released shortly that includes BlogSec’s recent addition, WP-Lockdown, so keep posted.

[6] More information about that file can be found here: http://httpd.apache.org/docs/1.3/mod/mod_auth.html

WordPress Plugin Tracker – Are you updated?

If you just installed your blog and all plugins from the developer website you should already be running the latest version. You should install the WordPress Plugin Tracker plugin to keep track if you're using the latest plugin versions. After you installed and activated the plugin, run it to see if your plugins are the current versions. You should see something like this:



Plugin Release Tracker

Track the releases of the plugins you have installed in your website

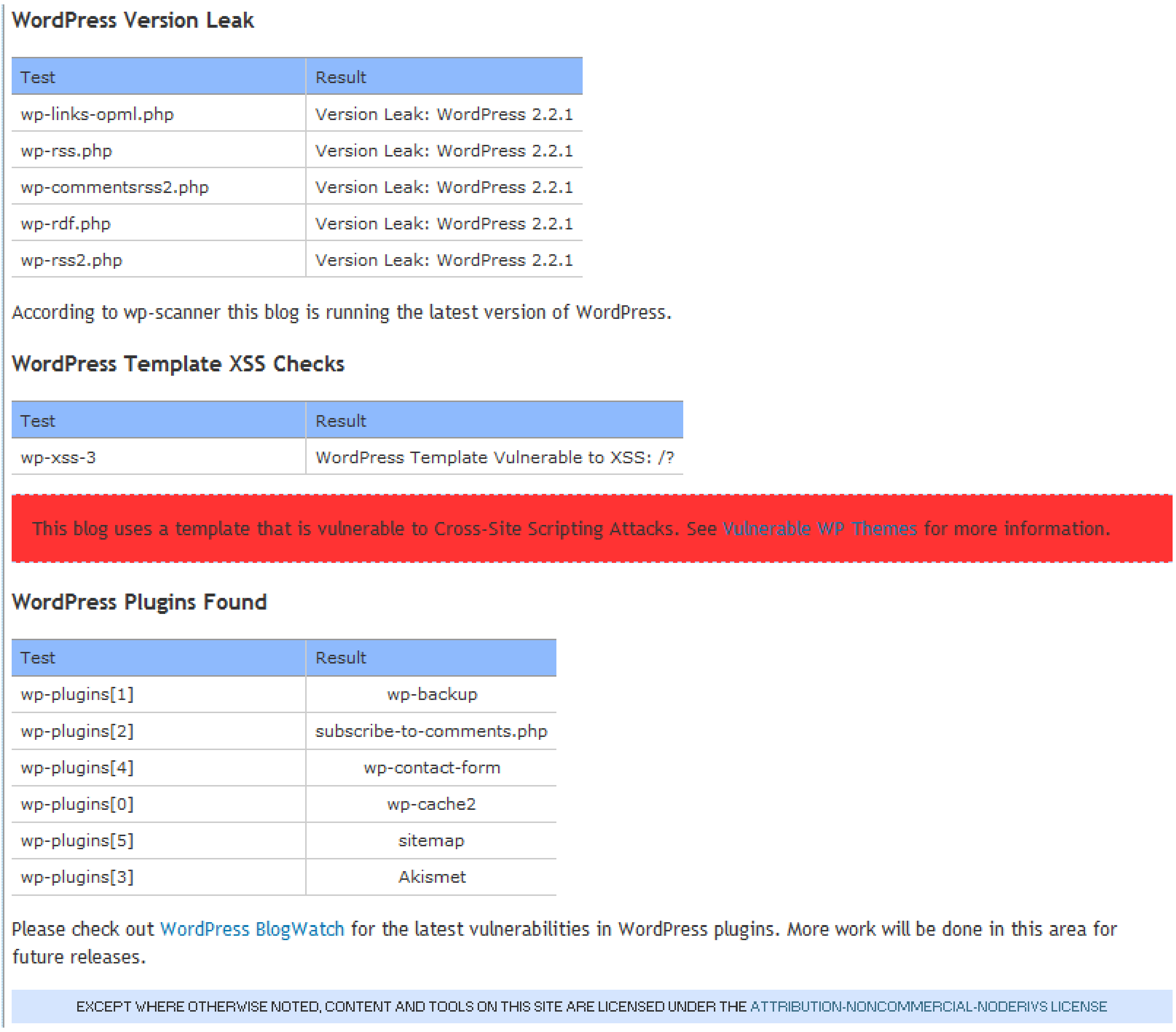
Plugin	Your Version	WPPDB Version	Status
Another Wordpress Meta Plugin	2.0.3	2.0.3	Versions are matching, You have latest v
Akismet	2.0.2	2.0.2	Versions are matching, You have latest v
Bad Behavior	2.0.10	2.0.10	Versions are matching, You have latest v
http:BL WordPress Plugin	1.4	1.4	Versions are matching, You have latest v

This is how the example page of the Plugin Release Tracker looks

If some plugins aren't up to date, you'll be notified about it and by clicking on the title of the plugin to the left, you're easily forwarded to the plugin page where you can grab the update. So it's easy to keep your blog updated.

WordPress Online Security Scanner

BlogSecurity has written a WordPress security checking tool, to check your blog for common security weaknesses. It is excellent at enumerating plugins, checking for Cross-Site Scripting vulnerabilities and much more.



WordPress Version Leak

Test	Result
wp-links-opml.php	Version Leak: WordPress 2.2.1
wp-rss.php	Version Leak: WordPress 2.2.1
wp-commentsrss2.php	Version Leak: WordPress 2.2.1
wp-rdf.php	Version Leak: WordPress 2.2.1
wp-rss2.php	Version Leak: WordPress 2.2.1

According to wp-scanner this blog is running the latest version of WordPress.

WordPress Template XSS Checks

Test	Result
wp-xss-3	WordPress Template Vulnerable to XSS: /?

This blog uses a template that is vulnerable to Cross-Site Scripting Attacks. See Vulnerable WP Themes for more information.

WordPress Plugins Found

Test	Result
wp-plugins[1]	wp-backup
wp-plugins[2]	subscribe-to-comments.php
wp-plugins[4]	wp-contact-form
wp-plugins[0]	wp-cache2
wp-plugins[5]	sitemap
wp-plugins[3]	Akismet

Please check out WordPress BlogWatch for the latest vulnerabilities in WordPress plugins. More work will be done in this area for future releases.

EXCEPT WHERE OTHERWISE NOTED, CONTENT AND TOOLS ON THIS SITE ARE LICENSED UNDER THE ATTRIBUTION-NONCOMMERCIAL-NODERIVS LICENSE

WP-Scanner is a free online service and has tested more then 5000 blogs to date. Check out more information here.

The End

We reached the end of our whitepaper. We hope you enjoyed it, and that you'll have success implementing these vital security steps. Its always a pleasure to get feedback and your stories so please contact us.