



The Ultimate WordPress Security Checklist

Who Am I?

- Working in Cloud & Application security from last 11 years
- Working as Technology Head of WordPress team @miniOrange

~ Kalpesh Hiran





Why WordPress Security Matters?

- WordPress Website WordPress is the most popular CMS globally, **powering over 43%** of the Internet
- Millions of people around the world use WordPress.
- Because of its popularity, the CMS is a **prime target for hackers** and malicious users



Isn't WordPress Secure By Default?

“Yes” and “No”

- WordPress is **well-maintained** platform, and the WordPress community work hard to keep WordPress core vulnerability free
- WordPress's few default settings may leave websites vulnerable to security threats
- Many security threats come from **themes, plugins, or other third-party software** that is added to the site or due to **security misconfigurations**



Security Practices to secure your WordPress

Strong Passwords



 Password 

 Yc4gwy8@ 



Why is it required?

- WordPress faces numerous **brute force attacks** where hackers attempt to guess passwords.
- Strong passwords significantly increase the complexity of guessing, making it harder for attackers to gain unauthorized access.

Many users still run their administrator account with “admin” as a password

A screenshot of the WordPress login interface. At the top center is the WordPress logo. Below it is a white login form with a light gray border. The form contains two input fields: the first is labeled 'Username or Email Address' and contains the text 'admin'; the second is labeled 'Password' and also contains 'admin', with a small blue eye icon to its right. Below the password field is a checkbox labeled 'Remember Me' which is unchecked. To the right of the checkbox is a blue 'Log In' button.



How to achieve?

- Enforce all users or specific roles to use strong passwords with password policy plugins
- Ask users to reset password on their first login
- Set expiry for passwords

Select the specific set of Policy Settings for your users.

For all Users Specific Roles

Password Policy Settings

Enable/Disable all settings

Policy Settings

Must Contain Lower and Uppercase letter like [a|A]

Must Contain Numeric digits like [0,9]

Must Contain characters like [@, #, \$, % etc]

Length of password [between 8 and 25]

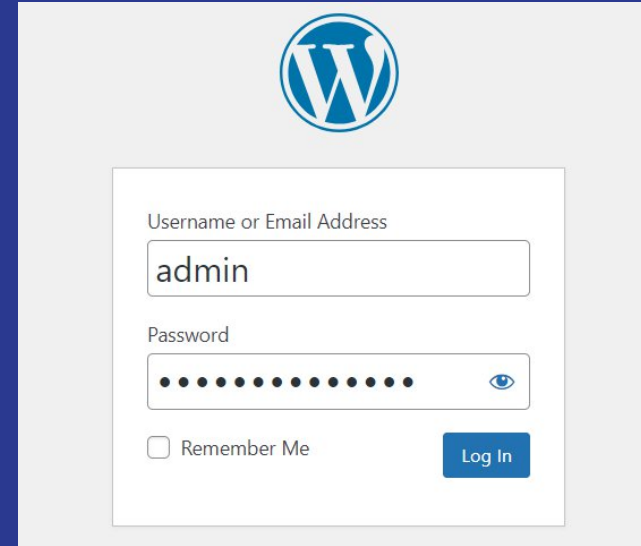
Force reset password on first login

Enable expiration time

Expiry Time

Password will be expired after 7 weeks

Change the Default “admin” username

A screenshot of the WordPress login page. At the top center is the WordPress logo. Below it is a white login form with a light gray border. The form contains the following elements: a text input field labeled 'Username or Email Address' with the text 'admin' entered; a password input field with a series of black dots and a blue eye icon for toggling visibility; a checkbox labeled 'Remember Me' which is currently unchecked; and a blue 'Log In' button on the right side of the form.



How to achieve?

- Create strong, unique usernames during **WordPress installation** or when creating new user accounts.
- Avoid using common usernames such as "admin", "administrator", or "root"
- If the default admin username is already in use, create a new administrator user with a different username and delete the default admin account.

Hiding default Login page



wp-login.php
not found



.....
WordPress



Why is it required?

- [wp-login.php](#) is the default login page for WordPress, making it a common target for brute force attacks.
- By hiding or renaming wp-login.php, you can mitigate the risk of direct brute force attacks on the login page.



How to achieve?

- Access your WordPress site's root directory via FTP or file manager
- Rename `wp-login.php` to a different, hard-to-guess name (e.g, `my-login.php`)
- Update any internal links or scripts referencing the login page to reflect the new URL

Limit Login Attempts





Why is it required?

- By default, WordPress allows users to try to **login as many time as they want**.
- This leaves your WordPress site vulnerable to brute force attacks.
- Hackers try to crack passwords by trying to login with different combinations

A screenshot of the WordPress login page. At the top left is the WordPress logo, a white 'W' inside a circle. To its right is the word "WORDPRESS" in a blue, serif font. Below the logo and text is a red-bordered box containing two error messages: "ERROR: Incorrect username or password." and "ERROR: Too many failed login attempts. Please try again in 20 minutes." Below the error box is a white login form. It has a "Username" label above a text input field containing the text "cnick". Below that is a "Password" label above an empty password input field. At the bottom left of the form is a checkbox labeled "Remember Me". At the bottom right is a blue button with the text "Log In" in white.



How to achieve?

- Install and activate a WordPress security plugin such as "**Limit Login Attempt**" or "**Wordfence Security**"
- Configure the plugin settings to limit the number of login attempts **allowed within a specified time frame**
- Optionally, set up email notifications or alerts to **notify administrators of suspicious login attempts**

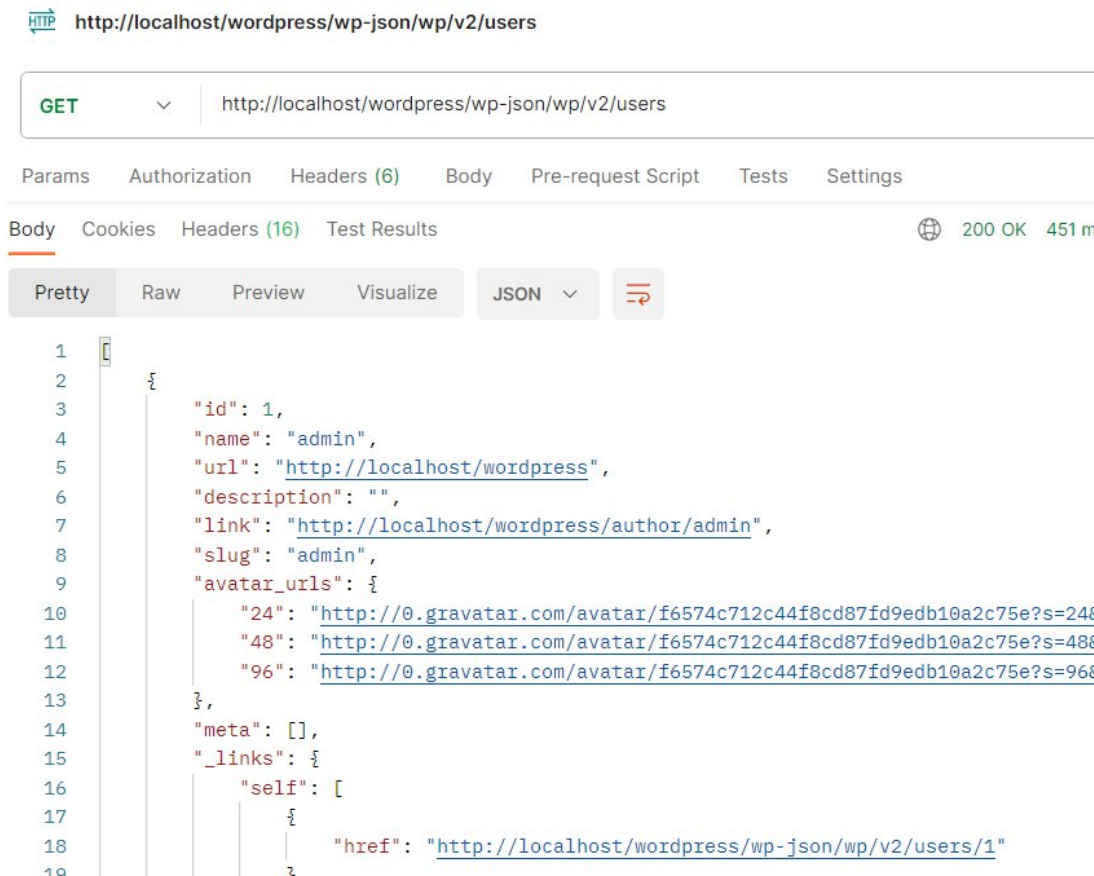


Disable WordPress REST APIs



Why is it required?

- WordPress APIs like [/wp-json/wp/v2/users](http://localhost/wordpress/wp-json/wp/v2/users) (which don't require any authentication) can expose sensitive user data and **expose usernames** of registered users on wordpress site



```
HTTP http://localhost/wordpress/wp-json/wp/v2/users

GET http://localhost/wordpress/wp-json/wp/v2/users

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Body Cookies Headers (16) Test Results 200 OK 451 m

Pretty Raw Preview Visualize JSON

1 {
2   "id": 1,
3   "name": "admin",
4   "url": "http://localhost/wordpress",
5   "description": "",
6   "link": "http://localhost/wordpress/author/admin",
7   "slug": "admin",
8   "avatar_urls": {
9     "24": "http://0.gravatar.com/avatar/f6574c712c44f8cd87fd9edb10a2c75e?s=24",
10    "48": "http://0.gravatar.com/avatar/f6574c712c44f8cd87fd9edb10a2c75e?s=48",
11    "96": "http://0.gravatar.com/avatar/f6574c712c44f8cd87fd9edb10a2c75e?s=96"
12  },
13  "meta": [],
14  "_links": {
15    "self": [
16      {
17        "href": "http://localhost/wordpress/wp-json/wp/v2/users/1"
18      }
19    ]
20  }
```



How to achieve?

- **Disable WordPress APIs** if you are not using it
- Implement **authentication** and **authorization** to restrict access to WordPress APIs
- Utilize either available plugins or custom code to achieve disabling your APIs or put them behind authentication

Two-Factor Authentication (2FA)





Why is it required?

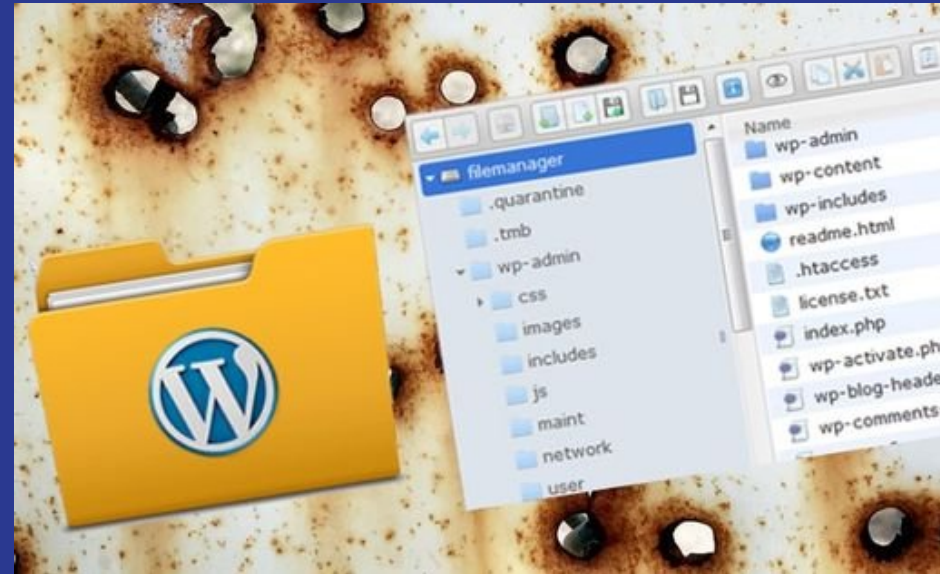
- Passwords alone may be compromised through various means, including **phishing attacks**.
- Two-factor authentication (2FA) adds an extra layer of security, requiring an additional verification step to access accounts like OTP, TOTP or Biometric



How to achieve?

- Enable two-factor authentication for all user accounts through WordPress security plugins like [WordPress Two Factor](#)
- Encourage users, especially **administrators**, to use **authenticator** apps (for time based tokens) or **SMS-based codes** for 2FA

Disable File Editing





Why is it required?

- WordPress comes with a built-in code editor which allows you to edit your theme and plugin files
- Allowing file editing in WordPress dashboard can pose a security risk, as it **provides attackers with direct access to critical files** on your server

A screenshot of the WordPress dashboard's Theme Editor interface. On the left is a dark sidebar menu with options: Dashboard, Posts, Media, Pages, Comments, Appearance (highlighted), Themes, Customize, Widgets, Menus, Theme Editor, Plugins, Users, Tools, and Settings. The main content area is titled "Edit Themes" and contains a "Did you know?" message. Below that, it shows the "Twenty Nineteen: Stylesheet (style.css)" file being edited. The "Selected file content:" section displays the following code:


```
1 #charset "UTF-8";
2 /*
3 Theme Name: Twenty Nineteen
4 Theme URI: https://github.com/WordPress/twentynineteen
5 Author: the WordPress team
6 Author URI: https://wordpress.org/
7 Description: Our 2019 default theme is designed to show off the power of the block ed
8 all the default blocks, and is built so that what you see in the editor looks like wh
9 Nineteen is designed to be adaptable to a wide range of websites, whether you're runn
10 business, or supporting a non-profit. Featuring ample whitespace and modern sans-seri
11 body text, it's built to be beautiful on all screen sizes.
12 Requires at least: WordPress 4.9.6
13 Version: 1.2
14 License: GNU General Public License v2 or later
15 License URI: LICENSE
16 Text Domain: twentynineteen
17 Tags: one-column, flexible-header, accessibility-ready, custom-colors, custom-menu, cu
```




How to achieve?

- Access your WordPress site's wp-config.php file via FTP or file manager.
- Add the following line of code to the file:

```
define('DISALLOW_FILE_EDIT', true);
```

 Copy code



Change WordPress Database Prefix

change wp_



Why is it required?

- The default database prefix used by WordPress is **"wp_"** which is widely known and can be targeted for **SQL injection attacks**
- With default database prefix, it makes it easier for hackers to **guess what your table name is**

<input type="checkbox"/>	wp_postmeta	★	Browse	Structure	Search
<input type="checkbox"/>	wp_posts	★	Browse	Structure	Search
<input type="checkbox"/>	wp_termmeta	★	Browse	Structure	Search
<input type="checkbox"/>	wp_terms	★	Browse	Structure	Search
<input type="checkbox"/>	wp_term_relationships	★	Browse	Structure	Search
<input type="checkbox"/>	wp_term_taxonomy	★	Browse	Structure	Search
<input type="checkbox"/>	wp_um_metadata	★	Browse	Structure	Search
<input type="checkbox"/>	wp_usermeta	★	Browse	Structure	Search
<input type="checkbox"/>	wp_users	★	Browse	Structure	Search



How to achieve?

- **For New Installation**, during your installation **choose DB prefix which is hard to guess**
- **For existing installations**, change table names manually with either **ALTER TABLE** or tools like **PHPMyAdmin** and replace same prefix value in your WordPress configuration file **(wp-config.php)**

(ensure you have a backup of your database if something goes wrong)



Disable Directory Indexing and Browsing

Index of /wp-includes

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 Parent Directory		-	
 ID3/	21-Dec-2014 14:05	-	
 SimplePie/	21-Dec-2014 14:05	-	
 Text/	21-Dec-2014 14:05	-	
 admin-bar.php	21-Dec-2014 14:17	25K	
 atomlib.php	24-Nov-2014 17:45	11K	
 author-template.php	21-Dec-2014 14:17	14K	
 bookmark-template.php	21-Dec-2014 14:17	11K	
 bookmark.php	21-Dec-2014 14:17	13K	
 cache.php	21-Dec-2014 14:17	19K	



Why is it required?

- Directory indexing and browsing allow anyone to view the contents of directories on your web server, potentially **exposing sensitive information** or files
- Directory browsing can be used by hackers to find out if you have any files with known vulnerabilities, so they can take advantage of these files to gain access.



How to Disable?

- Access your web server's configuration files (e.g. **.htaccess** for Apache servers)
- Add the following directives to disable directory indexing and browsing:

```
Options -Indexes
```

Copy code



Disable XML-RPC





Why is it required?

- XML-RPC was enabled by default in WordPress 3.5 because it helps connecting your WordPress site with web and mobile apps
- Traditionally if a hacker wanted to try 500 different passwords on your website, they would have to make 500 separate login attempts, but with XML-RPC, a hacker can use the **system.multicall** function to try thousands of password with say 20 or 50 requests
- Because of its powerful nature, XML-RPC can significantly amplify the brute-force attacks



How to Disable?

- Add the following code snippet to your theme's **functions.php** file to disable XML-RPC

php

Copy code

```
// Disable XML-RPC
add_filter('xmlrpc_enabled', '__return_false');
```

- Alternatively, install and activate the "**Disable XML-RPC**" plugin from the WordPress repository



Keep WordPress version updated

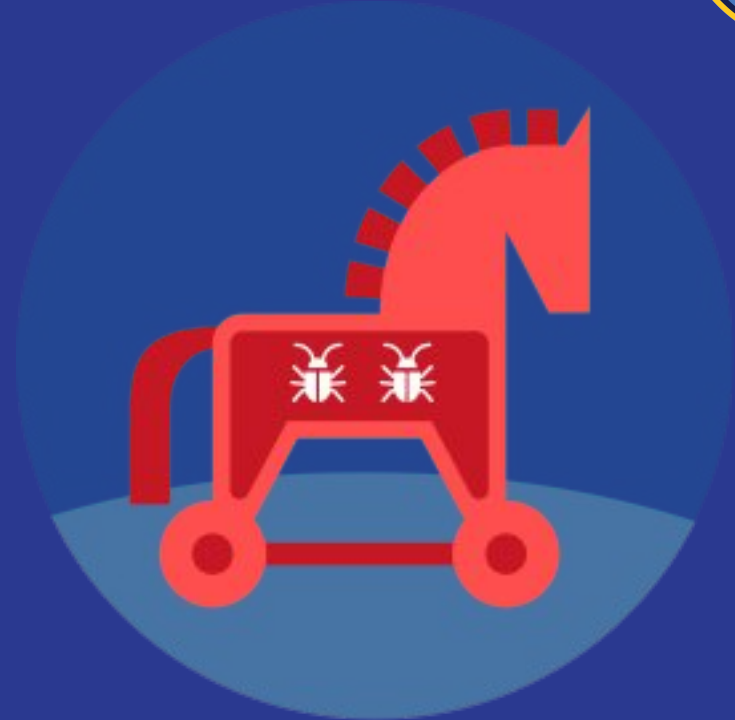
Running outdated versions of WordPress increases the risk of exploitation by malicious actors targeting known vulnerabilities.



How to achieve?

- Regularly **check for available updates** and apply them promptly through the WordPress dashboard
- Enable automatic updates for **WordPress core, themes, and plugins** to ensure timely installation of security patches.

Avoid Using Plugins & Themes from Untrusted Sources





How to achieve?

- Only download and install plugins and themes from trusted sources such as the **official WordPress repository** or reputable companies
- Verify the reputation and credibility of plugin and theme developers before installation by checking **reviews, ratings, and community feedback**



Key Takeaways

- Implement security measures such as **strong passwords** and **2-factor authentication** without relying solely on third-party solutions
- **Hide or rename WP defaults** (like **wp-login.php** or **wp_** database prefix) to mitigate common attack vectors
- Keep WordPress core, themes, and plugins **updated** to patch security vulnerabilities
- Avoid using plugins and themes from **untrusted sources**

Action Plan for Next **1 week**

- Switch to **strong** username and password
- Hide default **login page** & put **login limit**
- **Disable WP defaults** file editing, directory index, XMLRPC
- Disable **WordPress APIs**
- Enforce **2-factor** for Admin users



Action Plan for Next **1 month**

- Enforce 2FA for all users
- Put Authentication & Authorization on **WordPress APIs**
- Change default DB prefix “wp_”
- Update **WP core, plugins & themes** to latest one
- Identify & fix **plugins/themes with vulnerability** in active _____version





Q&A

Any questions?



Thank you!

Scan this QR and share your valuable feedback.

