

WORDPRESS FOR **PENTESTER**



LAB SETUP



Contents

Introduction	3
WordPress Setup on Ubuntu 20.04.....	3
Prerequisites for WordPress.....	3
Install Apache	3
Install MySQL.....	3
Install php.....	5
Create a Database for WordPress	6
WordPress Installation & Configuration.....	6
Install WordPress using Docker	13
Install WordPress on Windows Platform.....	18
WordPress Vulnerable Plugin	23



Introduction

In this post, we will demonstrate how to set up a vulnerable WordPress CMS for penetration testing on Ubuntu 20.04, using Docker and XAMPP on Windows.

WordPress Setup on Ubuntu 20.04

To configure WordPress on your Ubuntu platform, certain prerequisites are required for CMS installation.

Prerequisites for WordPress

- Apache
- Database (MySQL/MariaDB)
- PHP

Install Apache

Let's start the HTTP service with the help of Apache using a privileged account (as root), execute the following command in the terminal.

```
apt install apache2
```

```
root@ubuntu:~# apt install apache2 ←
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1 lib
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libapr
0 upgraded, 9 newly installed, 0 to remove and 198 not upgraded
Need to get 1,910 kB of archives.
```

Install MySQL

To run WordPress, you will also need a database server. The database server is where WordPress content is saved. So, we are going to choose MariaDB-server as the required database for WordPress and execute the following command

```
apt install mariadb-server mariadb-client
```



```
root@ubuntu:~# apt install mariadb-server mariadb-client ←
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  galera-3 gawk libaio1 libcgi-fast-perl libcgi-pm-perl libconfig-inifiles-
  libterm-readkey-perl mariadb-client-10.3 mariadb-client-core-10.3 mariadb
Suggested packages:
  gawk-doc libclone-perl libmldb-perl libnet-daemon-perl libsql-statement-
The following NEW packages will be installed:
  galera-3 gawk libaio1 libcgi-fast-perl libcgi-pm-perl libconfig-inifiles-
  libterm-readkey-perl mariadb-client mariadb-client-10.3 mariadb-client-co
0 upgraded, 22 newly installed, 0 to remove and 198 not upgraded.
Need to get 20.1 MB of software.
```

Next, execute the following commands to protect remote root login for the database server.

`mysql_secure_installation`

Then respond to questions asked after the command has been executed.

- Enter current password for root (enter for none): **press Enter**
- Set root password? [Y/n]: **Y**
- New password: **Enter password**
- Re-enter new password: **Repeat password**
- Remove anonymous users? [Y/n]: **Y**
- Disallow root login remotely? [Y/n]: **Y**
- Remove the test database and access to it? [Y/n]: **Y**
- Reload privilege tables now? [Y/n]: **Y**



```
root@ubuntu:~# mysql_secure_installation ←
NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
      SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
you haven't set the root password yet, the password will be blank,
so you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password ensures that nobody can log into the MariaDB
root user without the proper authorisation.

Set root password? [Y/n] Y ←
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
... Success!

By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them. This is intended only for testing, and to make the installation
go a bit smoother. You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] Y ←
... Success!

Normally, root should only be allowed to connect from 'localhost'. This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] Y ←
... Success!

By default, MariaDB comes with a database named 'test' that anyone can
access. This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n] Y ←
- Dropping test database...
... Success!
- Removing privileges on test database...
... Success!

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

Reload privilege tables now? [Y/n] Y ←
```

Install php

And at last, install the PHP MySQL and run the following command to install this application.

```
apt install php php-mysql
```



```
root@ubuntu:~# apt install php php-mysql ←
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libapache2-mod-php7.4 php-common php7.4 php7.4-cli php7.4-com
Suggested packages:
  php-pear
The following NEW packages will be installed:
  libapache2-mod-php7.4 php php-common php-mysql php7.4 php7.4-
0 upgraded, 11 newly installed, 0 to remove and 198 not upgrade
```

Create a Database for WordPress

To access MySQL, enter the following command, which will create a database for WordPress.

```
mysql -u root -p
CREATE DATABASE wordpress;
CREATE USER 'wp_user'@'localhost' IDENTIFIED BY 'password';
GRANT ALL ON wordpress.* TO 'wp_user'@'localhost' IDENTIFIED BY 'password';
FLUSH PRIVILEGES;
exit
```

```
root@ubuntu:~# mysql -u root -p ←
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 55
Server version: 10.3.22-MariaDB-1ubuntu1 Ubuntu 20.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE DATABASE wordpress; ←
Query OK, 1 row affected (0.000 sec)

MariaDB [(none)]> CREATE USER 'wp_user'@'localhost' IDENTIFIED BY 'password'; ←
Query OK, 0 rows affected (0.000 sec)

MariaDB [(none)]> GRANT ALL ON wordpress.* TO 'wp_user'@'localhost' IDENTIFIED BY 'password'; ←
Query OK, 0 rows affected (0.000 sec)

MariaDB [(none)]> FLUSH PRIVILEGES; ←
Query OK, 0 rows affected (0.000 sec)

MariaDB [(none)]> exit ←
Bye
root@ubuntu:~#
```

WordPress Installation & Configuration

Now, it's time to download and install WordPress on our localhost. With the help of the wget command, we have fetched the compressed file of WordPress setup and extracted the folder inside the /var/www/html directory.

```
cd /var/www/html
wget http://www.wordpress.org/latest.tar.gz
tar -xvf latest.tar.gz
```



```
root@ubuntu:/var/www/html# wget https://wordpress.org/latest.tar.gz ←
--2020-06-30 11:12:06-- https://wordpress.org/latest.tar.gz
Resolving wordpress.org (wordpress.org)... 198.143.164.252
Connecting to wordpress.org (wordpress.org)|198.143.164.252|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 12238031 (12M) [application/octet-stream]
Saving to: 'latest.tar.gz'

latest.tar.gz                                         100%[=====] 2020-06-30 11:12:43 (412 KB/s) - 'latest.tar.gz' saved [12238031/12238031]

root@ubuntu:/var/www/html# ls
index.html  latest.tar.gz
root@ubuntu:/var/www/html# tar -xvf latest.tar.gz ←
wordpress/
wordpress/xmlrpc.php
wordpress/wp-blog-header.php
wordpress/readme.html
wordpress/wp-signup.php
```

Then run the given command to change ownership of the 'wordpress' directory as well permission for the upload directory.

```
chown -R www-data:www-data wordpress/
chmod -R 755 wordpress/
mkdir wordpress/wp-content/uploads
chown -R www-data:www-data wordpress/wp-content/uploads
```

```
root@ubuntu:/var/www/html# chown -R www-data:www-data wordpress/ ←
root@ubuntu:/var/www/html# chmod -R 755 wordpress/ ←
root@ubuntu:/var/www/html# mkdir wordpress/wp-content/uploads ←
root@ubuntu:/var/www/html# chown -R www-data:www-data wordpress/wp-content/uploads/ ←
root@ubuntu:/var/www/html# [ ]
```

Now, we have completed the installation up to this point; to create a WordPress website, we need to access the application over the web browser on localhost by executing the following steps, and then complete the remaining installation process.

```
http://localhost/wordpress/
```

This will open the setup file and ask you to choose your preferred language. I select **English** and then press the **continue** button.



C

localhost/wordpress/wp-admin/setup-config.php



W English (United States)

Afrikaans
العربية
العربية المغربية
অসমীয়া
گۈئىشى آذربایجان
Azərbaycan dili
Беларуская мова
Български
বাংলা
ଶ୍ରୀମଦ୍
Bosanski
Català
Cebuano
Čeština
Cymraeg
Dansk
Deutsch (Sie)
Deutsch (Schweiz)
Deutsch
Deutsch (Schweiz, Du)
Deutsch (Österreich)
ଶ୍ରୀମଦ୍
Ελληνικά¹
English (South Africa)
English (Canada)

Continue

Read the given content and press Let's go to continue the activity.



Welcome to WordPress. Before getting started, we need some information on the database. You will need to know the following items before proceeding.

1. Database name
2. Database username
3. Database password
4. Database host
5. Table prefix (if you want to run more than one WordPress in a single database)

We're going to use this information to create a wp-config.php file. **If for any reason this automatic file creation doesn't work, don't worry. All this does is fill in the database information to a configuration file. You may also simply open wp-config-sample.php in a text editor, fill in your information, and save it as wp-config.php.** Need more help? [We got it.](#)

In all likelihood, these items were supplied to you by your Web Host. If you don't have this information, then you will need to contact them before you can continue. If you're all ready...

[Let's go!](#)

To continue the activity, we need to enter the required details that will help the application to connect with the database, thus it should be the same information that we have entered above at the time of database we have created for WordPress.



localhost/wordpress/wp-admin/setup-config.php?step=1

Below you should enter your database connection details. If you're not sure about these, contact your host.

Database Name The name of the database you want to use with WordPress.

Username Your database username.

Password Your database password.

Database Host You should be able to get this info from your web host, if localhost doesn't work.

Table Prefix If you want to run multiple WordPress installations in a single database, change this.

Submit

And if your above-given detail is correct, you will get the Installation page as we have here.

localhost/wordpress/wp-admin/setup-config.php?step=2

All right, sparky! You've made it through this part of the installation. WordPress can now communicate with your database. If you are ready, time now to...

Run the installation

Now, after that, it will ask you to enter details for your Website which you want to host using WordPress CMS as shown in the image below, and then finally click on the install Tab.

Note: The User and Password asked before the installation is referred to your Database information, and the username and password asked after installed are referred to your application (CMS).



Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Don't worry, you can always change these settings later.

Site Title Hacking Articles

Username admin

Password Ignite@987

Your Email abc@gmail.com

Search Engine Visibility Discourage search engines from indexing this site
It is up to search engines to honor this request.

Install WordPress

And once it is done, you will get the application login page where you must enter credentials to access the dashboard of your CMS.



The screenshot shows the WordPress login interface. The URL in the browser is `localhost/wordpress/wp-login.php?redirect_to=http%3A%2F%2Flocalhost%2Fword`. The page features the classic blue 'W' logo at the top. Below it is a login form with fields for 'Username or Email Address' containing 'admin', 'Password' (with masked input), and a 'Remember Me' checkbox. A 'Log In' button is to the right of the form. Below the form are links for 'Lost your password?' and '← Back to Hacking Articles'.

You will get the dashboard where you can write your content to be posted on the website.

The screenshot shows the WordPress dashboard. The URL in the browser is `localhost/wordpress/wp-admin/`. The top navigation bar shows 'Howdy, admin'. The dashboard has a sidebar with icons for posts, media, comments, users, and more. The main content area features a 'Welcome to WordPress!' message and 'Get Started' options like 'Customize Your Site' and 'Change your theme completely'. It also includes 'Next Steps' like 'Write your first blog post', 'Add an About page', 'Set up your homepage', and 'View your site'. A 'More Actions' section offers links to 'Manage widgets', 'Manage menus', 'Turn comments on or off', and 'Learn more about getting started'. At the bottom, there are sections for 'Site Health Status' (showing 'No information yet...') and 'Quick Draft' (with a 'Title' field).



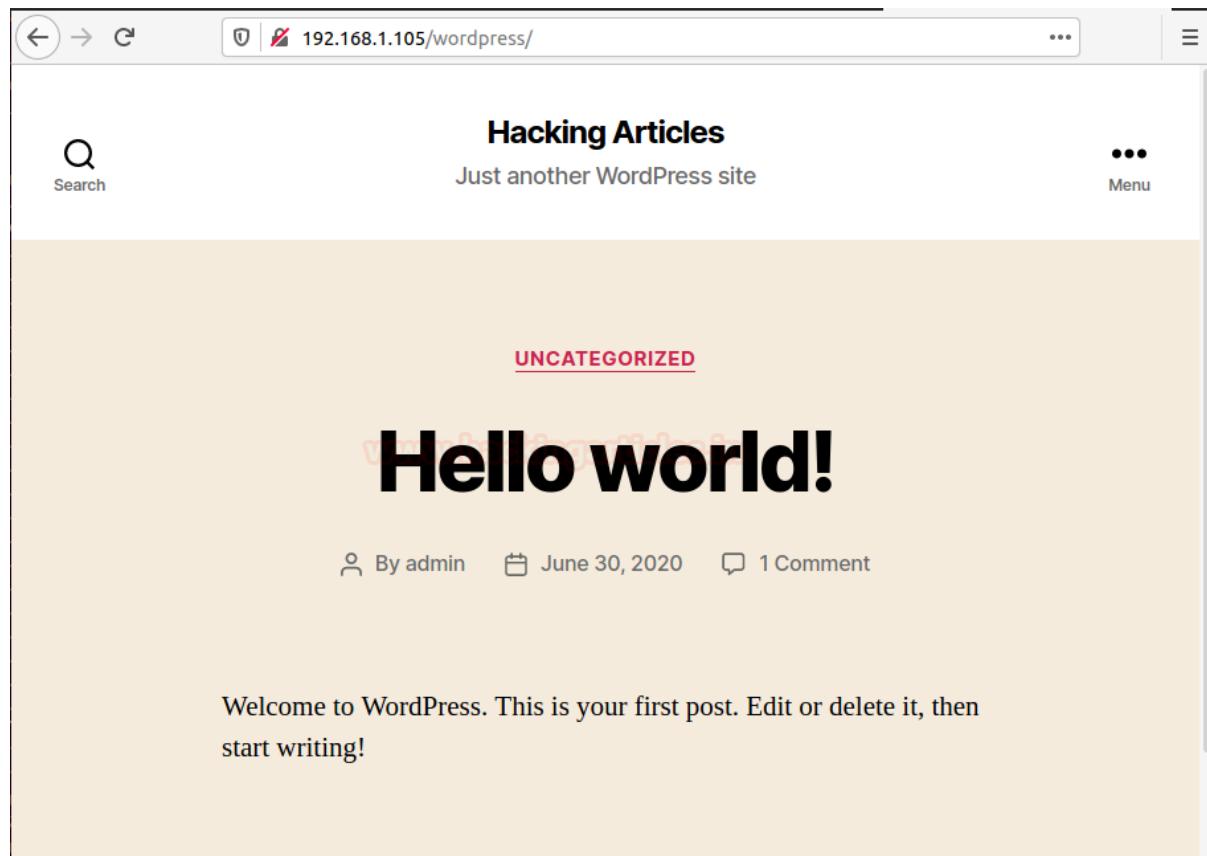
Open the wp-config.php file in WordPress directory and paste the following lines in it to access the website page.

```
define( 'WP_SITEURL', 'http://'. $_SERVER['HTTP_HOST'].'/wordpress');
define( 'WP_HOME', 'http://'. $_SERVER['HTTP_HOST'].'/wordpress');
```

```
/*
define( 'AUTH_KEY',         '5[F6RG{Txh-.(KwNW1<N-Owt3uW+$l.lovlz`7gU##C>8A7Mrecj4g>Jyu92zVF`' );
define( 'SECURE_AUTH_KEY',   'LY#0lh}3z]qQN@EC8kRLT_()PLR+`Cvv%vBC1l9LEo4i:%!axGLNTtM:pr_sAR^' );
define( 'LOGGED_IN_KEY',     'Gp#mWVZN8f$wkys/[Z(5KazPw)6=$z=d!=>FKo.!KeY0w-KeLPF:jU?e5o*R.w<>' );
define( 'NONCE_KEY',        '@Itsf#`7$an?-HXSG/=1]bLoR{!j^825R.InYzmyURV&yuXVeGb29Q:ZorT1[-Q' );
define( 'AUTH_SALT',         'VEK%c>$2LSw;(*>6w[^|nX|U)B!ml|=cDv?<T37xe]J&ZSGIvb92aN|J.#4p;z_i' );
define( 'SECURE_AUTH_SALT',  '2-$Y~#rI1U_0H3VdkZH+>9W}@oKUIE}~4qbE]_nFxZ6e5w|NeW?wdH^H;!iXmfR' );
define( 'LOGGED_IN_SALT',    '0qdVx)QH*Y=kF$wML>H]GHS{]G TePSy@.0kc_F-75qM>X{_N^R_U8:l!i(ou-xW' );
define( 'NONCE_SALT',        '0g}/<IKaxjn@<a)CRv1yRaBg6uRi=Vt.p%iT*:o(KLAY0sP6C>w}1}p35AFP$n[' );
define( 'WP_SITEURL',       'http://'. $_SERVER['HTTP_HOST'].'/wordpress');
define( 'WP_HOME',          'http://'. $_SERVER['HTTP_HOST'].'/wordpress');

/**#@-*/
/*
 * WordPress Database Table prefix.
 *
 * You can have multiple installations in one database if you give each
 * a unique prefix. Only numbers, letters, and underscores please!
 */
```

And finally, it is over here, and your WordPress is completely ready to go 😊 .



Install WordPress using Docker

Installing WordPress through Docker will release your effort of installing prerequisites for WordPress setup. It is a very easy and quick technique to configured WordPress. All you need to have some basic knowledge of Docker and its functionalities.



To install wordpress using docker, first, we will update the Ubuntu repository and then install the latest version of docker.io. Let's start the installation of docker packages with the apt command as below:

```
apt install docker.io
```

```
root@ubuntu:~# apt install docker.io ←
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  bridge-utils cgroupfs-mount containerd git git-man liberror-perl pigz
Suggested packages:
  ifupdown aufs-tools btrfs-progs debootstrap docker-doc rinse zfs-fuse
The following NEW packages will be installed:
  bridge-utils cgroupfs-mount containerd docker.io git git-man liberro
0 upgraded, 10 newly installed, 0 to remove and 198 not upgraded.
Need to get 74.8 MB of archives.
After this operation, 372 MB of additional disk space will be used.
```

Docker Compose is used to run multiple containers as a single service. Let's begin the installation of docker-compose with the help of apt by entering the following command.

```
apt install docker-compose
```

```
root@ubuntu:~# apt install docker-compose ←
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  python3-attr python3-cached-property python3-distutils python3-setuptools
  python3-texttable python3-websocket python3-xdg
Suggested packages:
  python-attr-doc python-jsonschema-doc python-setuptools-doc
The following NEW packages will be installed:
  docker-compose python3-attr python3-cached-property python3-distutils
  python3-persistent python3-setuptools python3-texttable python3-xdg
0 upgraded, 15 newly installed, 0 to remove and 198 not upgraded.
```

After installing the composer for the Docker, we must create a directory by the name of Wordpress. After creating the directory, we will create a .yml file that will contain the service definitions for your setup.

```
mkdir wordpress
cd wordpress/
nano docker-compose.yml
```

```
root@ubuntu:~# mkdir wordpress ←
root@ubuntu:~# cd wordpress/ ←
root@ubuntu:~/wordpress# nano docker-compose.yml ←
```



Now, paste the following text in the .yml and save the configuration. Source Code From [here](#)

```
version: '3.3' services:  
db:  
image: mysql:5.7  
volumes:  
- db_data:/var/lib/mysql  
restart: always  
environment:  
MYSQL_ROOT_PASSWORD: somewordpress  
MYSQL_DATABASE: wordpress  
MYSQL_USER: wordpress  
MYSQL_PASSWORD: wordpress  
  
wordpress:  
depends_on:  
- db  
image: wordpress:latest  
ports:  
- "8000:80"  
restart: always  
environment:  
WORDPRESS_DB_HOST: db:3306  
WORDPRESS_DB_USER: wordpress  
WORDPRESS_DB_PASSWORD: wordpress  
WORDPRESS_DB_NAME: wordpress  
volumes:  
db_data: {}
```



```
GNU nano 4.8
version: '3.3'

services:
  db:
    image: mysql:5.7
    volumes:
      - db_data:/var/lib/mysql
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: somewordpress
      MYSQL_DATABASE: wordpress
      MYSQL_USER: wordpress
      MYSQL_PASSWORD: wordpress

  wordpress:
    depends_on:
      - db
    image: wordpress:latest
    ports:
      - "8000:80"
    restart: always
    environment:
      WORDPRESS_DB_HOST: db:3306
      WORDPRESS_DB_USER: wordpress
      WORDPRESS_DB_PASSWORD: wordpress
      WORDPRESS_DB_NAME: wordpress
volumes:
  db_data: {}
```

Now run the Docker image in detached mode using the following command

```
docker-compose up -d
```

```
root@ubuntu:~/wordpress# docker-compose up -d ←
Creating network "wordpress_default" with the default driver
Creating volume "wordpress_db_data" with default driver
Pulling db (mysql:5.7)...
5.7: Pulling from library/mysql
8559a31e96f4: Downloading [=====>]
d51ce1c2e575: Download complete
c2344adc4858: Download complete
fcf3ceff18fc: Download complete
16da0c38dc5b: Download complete
b905d1797e97: Downloading [=====>]
4b50d1c6b05c: Download complete
d85174a87144: Download complete
a4ad33703fa8: Downloading [==>]
f7a5433ce20d: Waiting
3cd2a278b4a: Waiting
```



After doing all the configuration step-by-step, now access the localhost on port 8000 that will be hosting your WordPress Docker image and configure your WordPress site as done in the previous section.

The screenshot shows a browser window displaying the WordPress installation page at `localhost:8000/wp-admin/install.php?step=1`. The address bar is highlighted with a red box. The page title is "Welcome" and the subtext reads: "Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world." Below this, a section titled "Information needed" asks for site details. The "Site Title" field contains "Hacking Articles". The "Username" field contains "admin", with a note below stating: "Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol." The "Password" field contains "Ignite@987", with a "Hide" link and a "Medium" strength indicator. A note below says: "Important: You will need this password to log in. Please store it in a secure location." The "Your Email" field contains "abc@gmail.com", with a note: "Double-check your email address before continuing." Under "Search Engine Visibility", there is a checkbox labeled "Discourage search engines from indexing this site" which is unchecked, with a note: "It is up to search engines to honor this request." At the bottom is a blue "Install WordPress" button.

You will get the dashboard where you can write your content that to be posted on the website. But here we need to make some changes inside the **setting** so that the wordpress after installation it will work properly. Thus, enter your localhost IP address with a port number on which your docker image is running.



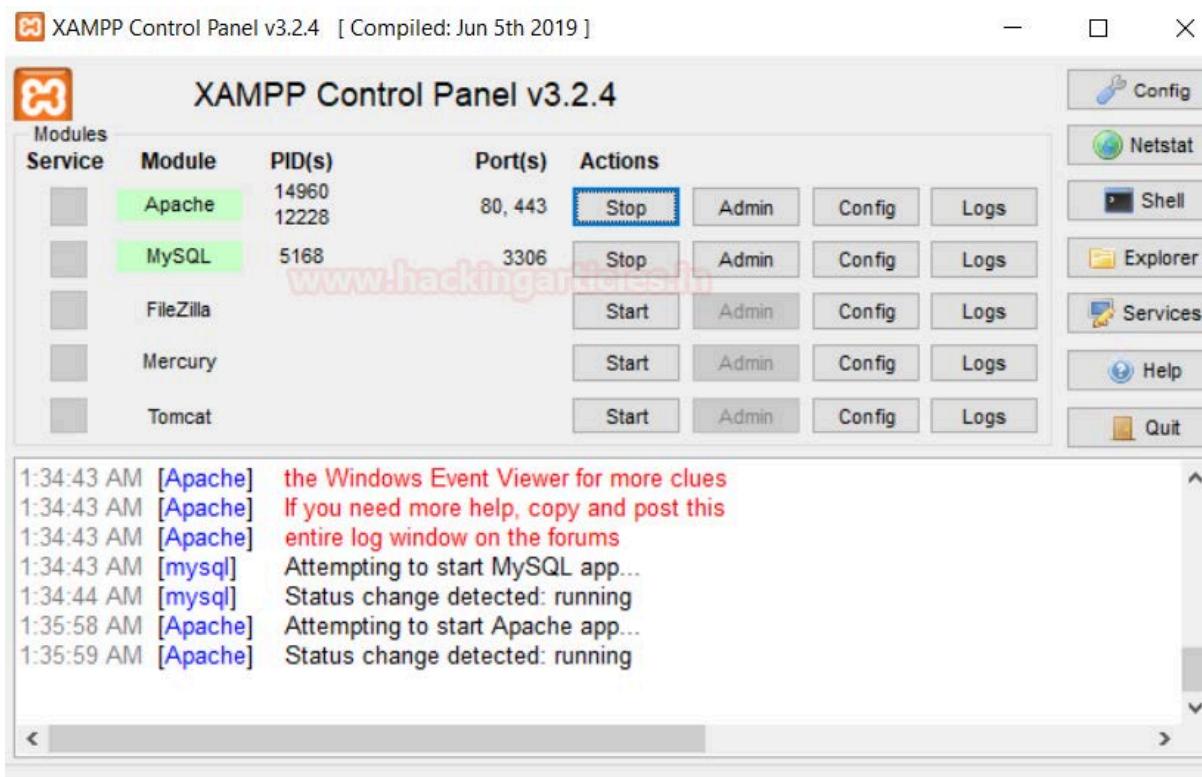
The screenshot shows the 'General Settings' page in the WordPress admin interface. The left sidebar is dark with white icons and text, showing 'Dashboard', 'Posts', 'Media', 'Pages', 'Comments', 'Appearance', 'Plugins (1)', 'Users', 'Tools', and 'Settings'. The 'Settings' option is highlighted with a blue bar. Under 'General', the 'Site Title' is set to 'Hacking Articles', the 'Tagline' is 'Just another WordPress site', and the 'WordPress Address (URL)' and 'Site Address (URL)' are both set to 'http://192.168.1.105:8000'. The 'Administration Email Address' is 'abc@gmail.com'. A checkbox for 'Anyone can register' is unchecked. The right side of the screen displays the configuration fields with their respective values.

And finally, it is over here, and your WordPress is completely ready to go but over port 8000 as shown here 😊 .

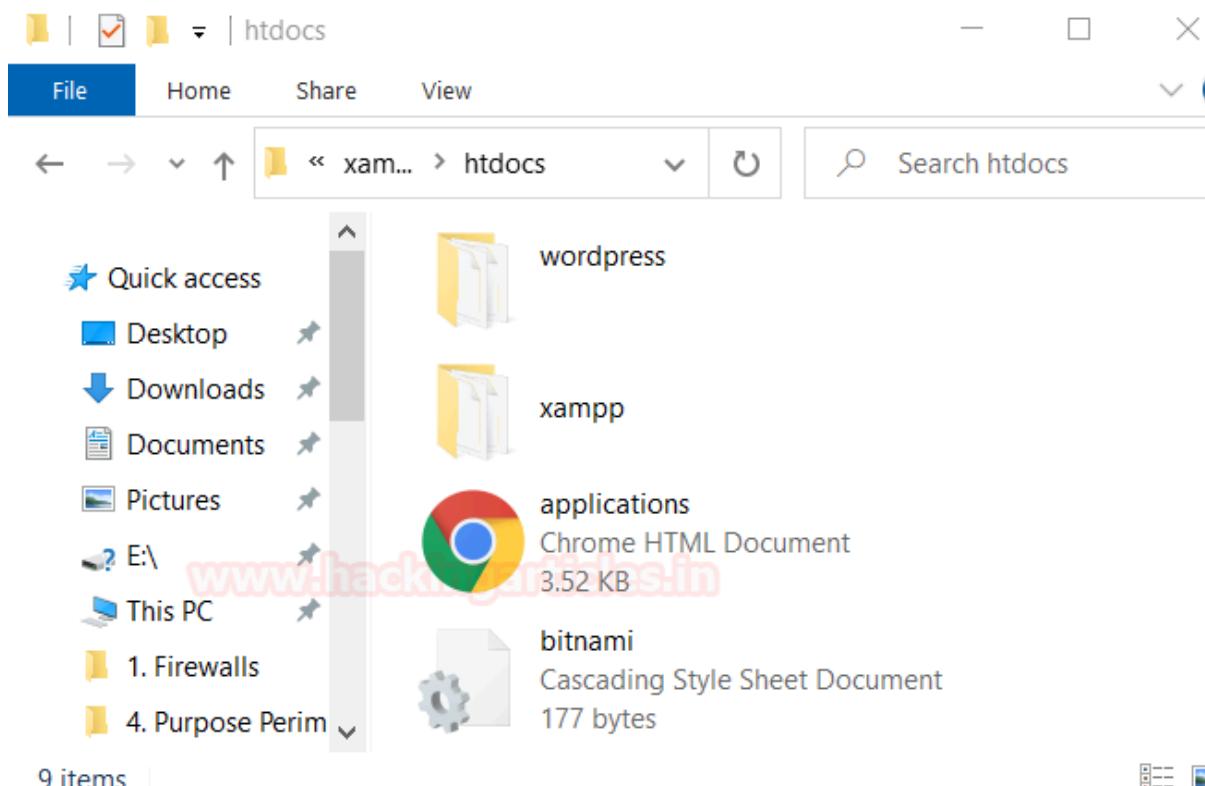
The screenshot shows the WordPress homepage. The header includes a search bar, the site title 'Hacking Articles', the tagline 'Just another WordPress site', and a menu icon. The main content features a large heading 'Hello world!' under the category 'UNCATEGORIZED'. Below the heading, it says 'By admin June 30, 2020 1 Comment'. The background of the page is light beige.

Install WordPress on Windows Platform

Installation of WordPress is also very easy as compared to Ubuntu because to fulfil the prerequisites of LAMP Server, we can use XAMPP, which will complete all the required dependencies like Apache and MySQL for WordPress.



Now download the extracted zip file of WordPress inside the /htdocs folder in /xampp folder in C-Drive.



Then, open the PHPMYADMIN in a web browser by accessing /localhost/phpMyAdmin and create the database for Wordpress to store its data.



The screenshot shows the phpMyAdmin interface for MySQL. The left sidebar lists databases: information_schema, mysql, performance_schema, phpmyadmin, and test. A 'New' icon is available for creating a new database. The main panel shows the 'Databases' table with the following data:

Database	Collation	Action
information_schema	utf8_general_ci	Check privileges
mysql	utf8mb4_general_ci	Check privileges
performance_schema	utf8_general_ci	Check privileges
phpmyadmin	utf8_bin	Check privileges
test	latin1_swedish_ci	Check privileges

Total: 5

Below the table are buttons for 'Check all' and 'With selected: Drop'. A note at the bottom reads: **Note:** Enabling the database statistics here might cause heavy traffic between the web server and the MySQL server.

Now to configure wordpress, explore the /localhost/wordpress/ and then enter the detail for the database.

Note: By Default, XAMPP DB_User is root and DB_Pass is empty <blank>

So as per XMAPP database configuration, we entered the following details in the given record.



Below you should enter your database connection details. If you're not sure about these, contact your host.

Database Name	<input type="text" value="wordpress"/> <small>www.hackingarticles.in</small>	The name of the database you want to use with WordPress.
Username	<input type="text" value="root"/>	Your database username.
Password	<input type="password"/>	Your database password.
Database Host	<input type="text" value="localhost"/>	You should be able to get this info from your web host, if localhost doesn't work.
Table Prefix	<input type="text" value="wp_"/>	If you want to run multiple WordPress installations in a single database, change this.

Now again repeat the same step as done in the above section.



Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Don't worry, you can always change these settings later.

Site Title Hacking Articles
www.hackingarticles.in

Username admin
Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.

Password Ignite@987
Medium
Hide

Important: You will need this password to log in. Please store it in a secure location.

Your Email abc@gmail.com
Double-check your email address before continuing.

Search Engine Visibility Discourage search engines from indexing this site
It is up to search engines to honor this request.

[Install WordPress](#)

You will get the dashboard where you can write your content that to be posted on the website.



The screenshot shows the WordPress dashboard. At the top, there's a header with the title "Dashboard < Hacking Articles" and a URL "localhost/wordpress/wp-admin/". Below the header is a navigation bar with links like "Hacking Articles", "3", "0", "+ New", "Howdy, admin", "Screen Options", and "Help". The main content area has a "Welcome to WordPress!" message and a "Get Started" section with a blue button "Customize Your Site" and a link "or, [change your theme completely](#)". To the right are sections for "Next Steps" (Write your first blog post, Add an About page, Set up your homepage, View your site) and "More Actions" (Manage widgets, Manage menus, Turn comments on or off, Learn more about getting started). A "Site Health Status" box says "No information yet..." and a "Quick Draft" box has fields for "Title" and "Content".

To make it vulnerable WordPress platform in order to perform penetration testing I have installed some vulnerable plugin as highlighted in the image.

To know how we can go do WordPress Penetration testing, read [this](#) article.

WordPress Vulnerable Plugin

- <https://www.exploit-db.com/exploits/40290>
- <https://www.exploit-db.com/exploits/36374>
- <https://www.exploit-db.com/exploits/44883>



localhost/wordpress/wp-admin/plugins.php?plugin_status=all&paged=1&s

All (6) | Active (4) | Inactive (2) | Update Available (3)

Bulk Actions

Plugin	Description
Akismet Anti-Spam	Used by millions, Akismet is quite possibly the best way in the world to protect your blog from spam. It keeps track of your API key.
Duplicator	Migrate and backup a copy of your WordPress files and database. Duplicate and move a site from one location to another.
Hello Dolly	This is not just a plugin, it symbolizes the hope and enthusiasm of an entire generation summed up in two words: Hello Dolly.
Mail Masta	Mail Masta is email marketing plugin for Wordpress.
ReFlex Gallery	Wordpress Plugin for creating responsive image galleries. By: HahnCreativeGroup
WP Google Maps	The easiest to use Google Maps plugin! Create custom Google Maps with high quality markers containing location supplied shortcode. No fuss.

Version 4.1.5 | By Automatic | View details

There is a new version of Akismet Anti-Spam available. [View version 4.1.6 details](#) or [update now](#).

Version 1.2.32 | By Snap Creek | View details | Go Pro

There is a new version of Duplicator available. [View version 1.3.36 details](#) or [update now](#).

Version 1.7.2 | By Matt Mullenweg | View details

Version 1.0 | By Mail Masta

Version 3.1.7 | By HahnCreativeGroup | View details

Version 3.4 | By WP Google Maps | View details

There is a new version of WP Google Maps available. [View version 8.0.25 details](#) or [update now](#).

www.hackingarticles.in

JOIN OUR TRAINING PROGRAMS

CLICK HERE

BEGINNER

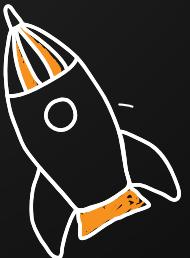
Ethical Hacking

Bug Bounty

Network Security Essentials

Network Pentest

Wireless Pentest



ADVANCED

Burp Suite Pro

Web Services-API

Pro Infrastructure VAPT

Computer Forensics

Android Pentest

Advanced Metasploit

CTF



EXPERT

Red Team Operation

Privilege Escalation

- APT's - MITRE Attack Tactics
- Active Directory Attack
- MSSQL Security Assessment

Windows

Linux

