DEPLOYING WORDPRESS ON UBUNTU SERVER AND MONITORING WEBSITE LOGS

REPORT PREPARED FOR

REDTEAM®
HACKER ACADEMY

Report issued

25/12/2024

Submitted By

SAJITH KUMAR S

TABLE OF CONTENTS

ABSTRACT

This study delves into the deployment of WordPress on an Ubuntu server and the incorporation of Splunk for monitoring and analyzing server logs, aiming to achieve superior performance, reliability, and security. WordPress, a widely utilized content management system, is installed and configured on an Ubutu server by setting up a robust software stack comprising Apache, PHP, and MySQL. The deployment process entails meticulous configuration of the server environment, ensuring seamless compatibility between the software components, and implementing security measures to safeguard the installation against potential vulnerabilities and threats.

To maintain the operational reliability of the WordPress site, effective monitoring is paramount. Splunk, a versatile and powerful platform for log management and analytics, is integrated into the server environment. Its deployment enables the collection, indexing, and in-depth analysis of logs generated by the server and the WordPress application. This integration provides actionable insights into key areas, including server performance, application errors, user activity patterns, and potential security risks. The study outlines the process of configuring log forwarding from the Ubuntu server to Splunk, designing dynamic dashboards for visualizing log data, and establishing alert mechanisms to promptly address emerging issues.

By combining the strengths of WordPress as a flexible web application platform and Splunk's advanced log analysis capabilities, this approach creates a comprehensive framework for hosting scalable, secure, and well-monitored web applications. The paper concludes by presenting best practices and strategic recommendations for maintaining system integrity, optimizing operational efficiency, and proactively addressing challenges through effective log monitoring and management.

ACKNOWLEDGEMENT

I wish to extend my heartfelt gratitude to all those who contributed to the successful completion of this project, Deploying WordPress on Ubuntu Server and Monitoring Website Logs. This achievement would not have been possible without the guidance, support, and collaboration of many individuals and organizations.

First and foremost, I am deeply thankful to Aswanarayan sir, whose invaluable expertise, encouragement, and guidance provided the foundation for this work. Their thoughtful advice and constructive feedback were instrumental in shaping both the direction and depth of this project, inspiring me to aim higher at every step.

I am equally grateful to Red Team Hacker Academy, Thiruvananthapuram, for providing the infrastructure, resources, and environment necessary for this project. A special note of appreciation goes to the Red Team Academy for their unwavering support in resolving technical challenges during the deployment and monitoring phases. Their professionalism and problem-solving skills were crucial to overcoming the obstacles encountered along the way.

I also want to express my sincere appreciation to the developers and maintainers of the open-source tools and platforms that formed the backbone of this project. The seamless integration of WordPress, Ubuntu, Apache, MySQL, PHP, and various log monitoring solutions was only possible due to their tireless efforts in advancing open-source technology.

On a personal note, I am immensely thankful to my family, friends, and colleagues for their constant encouragement, understanding, and patience throughout this journey. Their emotional support and motivational words were a source of strength during moments of challenge and self-doubt. This project stands as a testament to the power of collaboration, shared knowledge, and the spirit of continuous learning. I am profoundly grateful to everyone who played a role, directly or indirectly, in helping me bring this endeavor to fruition.

INTRODUCTION

Building a website can be an exciting venture, but ensuring it runs smoothly and remains secure is crucial. This guide will walk you through three essential tasks to get your WordPress site up and running on Ubuntu 24.04.1 LTS, enhance its security, and effectively monitor its performance with Splunk.

1. **Installing WordPress on Ubuntu24.04.1LTS**
   We'll start by setting up a WordPress website on Ubuntu 24.04.1 LTS, covering all the necessary steps, from configuring the server environment to getting WordPress up and running.

2. **Securing the WordPress Login Page**
   After installation, the next step is to secure your WordPress login page. It's critical to protect this entry point from attacks like brute force attempts. We'll explore methods for securing the login page and preventing unauthorized access.

3. **Monitoring Your Website with Splunk**
   Once your website is live, it's important to track its activity to ensure optimal performance and quickly address any issues. Splunk serves as a powerful tool for real-time log monitoring, helping you understand user behavior, identify errors, and improve the security of your website.

3.1 Install Ubuntu Server on VirtualBox

Download Ubuntu Server from the official site:

https://ubuntu.com/download/server

Install it on VirtualBox following the setup wizard.

Update System Packages

Run the following commands to update your system:

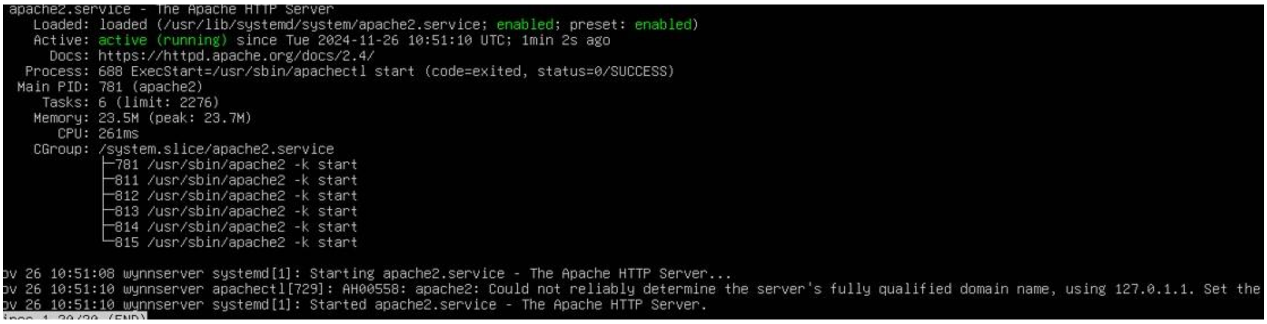
$ sudo apt update && apt upgrade

3.2 Install Apache

To install Apache web server:

$ sudo apt install apache2

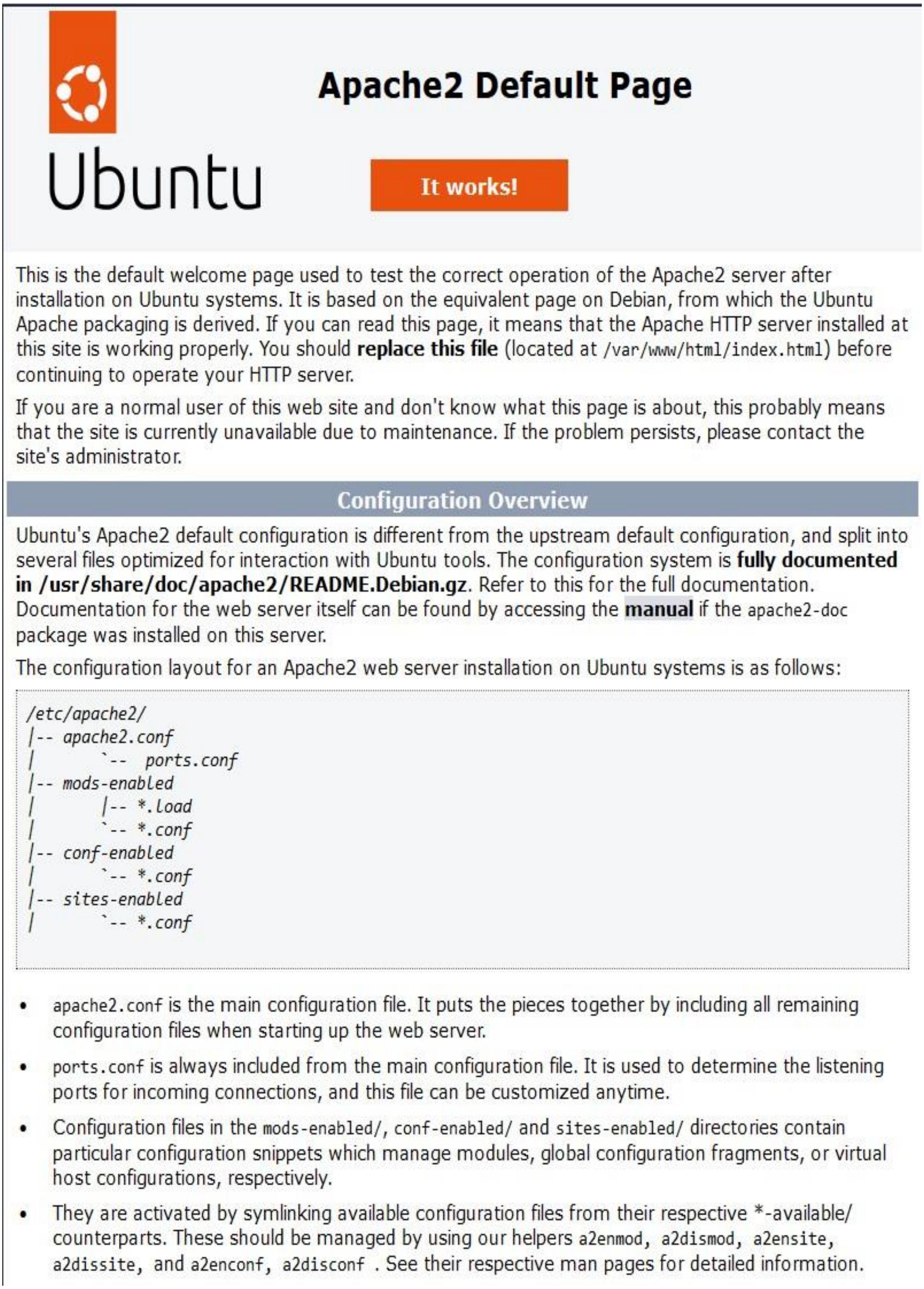
“ To check the status of Apache in our system, execute the following command “ **$** *sudo systemctl status apache2*



```
apache2.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
     Active: active (running) since Tue 2024-11-26 10:51:10 UTC; 1min 2s ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 688 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 781 (apache2)
      Tasks: 6 (limit: 2276)
     Memory: 23.5M (peak: 23.7M)
        CPU: 261ms
     CGroup: /system.slice/apache2.service
             ├─781 /usr/sbin/apache2 -k start
             ├─811 /usr/sbin/apache2 -k start
             ├─812 /usr/sbin/apache2 -k start
             ├─813 /usr/sbin/apache2 -k start
             ├─814 /usr/sbin/apache2 -k start
             └─815 /usr/sbin/apache2 -k start

ov 26 10:51:08 wynnserver systemd[1]: Starting apache2.service - The Apache HTTP Server...
ov 26 10:51:10 wynnserver apachectl[729]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the
ov 26 10:51:10 wynnserver systemd[1]: Started apache2.service - The Apache HTTP Server.
```

Access the server using your browser with the server's IP address

➔ **http://ip-address**



Apache2 Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

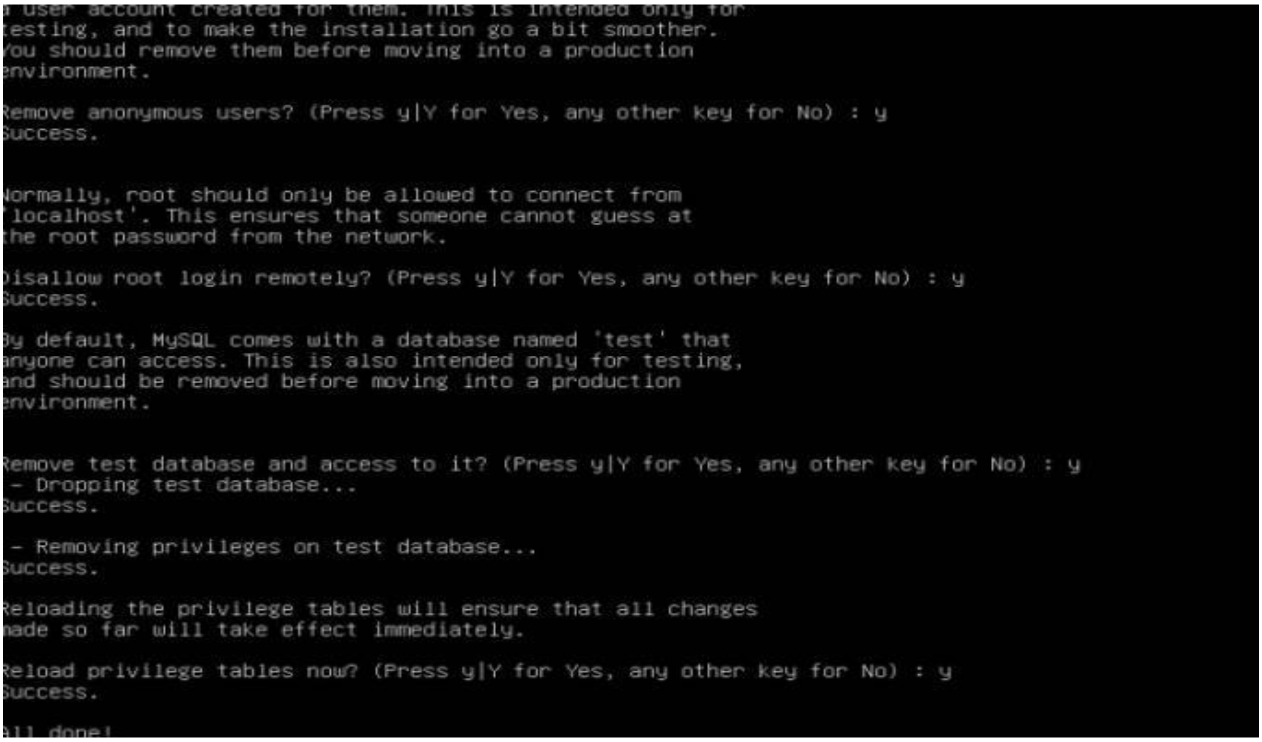
- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective *-available/ counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`, `a2dissite`, and `a2enconf`, `a2disconf` . See their respective man pages for detailed information.

INSTALL MySQL SERVER

$ *sudo apt install mysql-server mysql-client*

Secure MySQL Installation:

$ *sudo mysql_secure_installation*



```
a user account created for them. This is intended only for
esting, and to make the installation go a bit smoother.
ou should remove them before moving into a production
environment.

Remove anonymous users? (Press y|Y for Yes, any other key for No) : y
Success.


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

Disallow root login remotely? (Press y|Y for Yes, any other key for No) : y
Success.

By default, MySQL 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? (Press y|Y for Yes, any other key for No) : 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? (Press y|Y for Yes, any other key for No) : y
Success.

All done!
```

- **Remove anonymous users?** (y)
- **Disallow root login remotely?** (y)
- **Remove test database?** (y)
- **Reload privilege tables?** (y)

That's it, installation is secured!

INSTALL PHP

Install PHP and the required module

$ *sudo apt install php php-mysql*

"To confirm that PHP is installed"

Create an *"info.php" file at "/var/www/html/"*

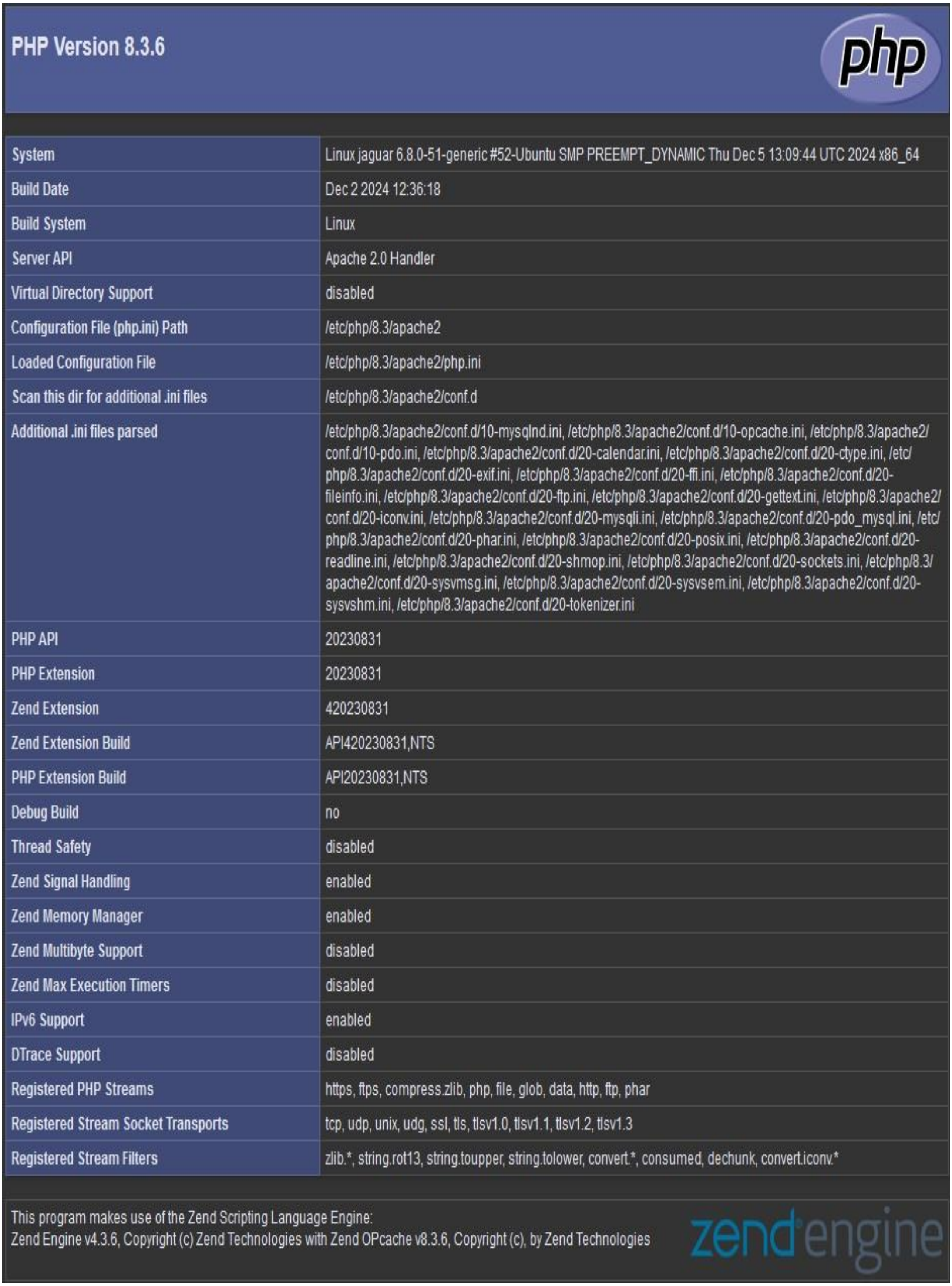
Path : **$** *sudo nano /var/www/html/info.php*

Add the following lines:

<?php phpinfo();

?>

Access the server using your browser with the server's IP address and /info.php --→ https://ip-address/info.php



PHP Version 8.3.6

System	Linux jaguar 6.8.0-51-generic #52-Ubuntu SMP PREEMPT_DYNAMIC Thu Dec 5 13:09:44 UTC 2024 x86_64
Build Date	Dec 2 2024 12:36:18
Build System	Linux
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/8.3/apache2
Loaded Configuration File	/etc/php/8.3/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/8.3/apache2/conf.d
Additional .ini files parsed	/etc/php/8.3/apache2/conf.d/10-mysqlnd.ini, /etc/php/8.3/apache2/conf.d/10-opcache.ini, /etc/php/8.3/apache2/conf.d/10-pdo.ini, /etc/php/8.3/apache2/conf.d/20-calendar.ini, /etc/php/8.3/apache2/conf.d/20-ctype.ini, /etc/php/8.3/apache2/conf.d/20-exif.ini, /etc/php/8.3/apache2/conf.d/20-ffi.ini, /etc/php/8.3/apache2/conf.d/20-fileinfo.ini, /etc/php/8.3/apache2/conf.d/20-ftp.ini, /etc/php/8.3/apache2/conf.d/20-gettext.ini, /etc/php/8.3/apache2/conf.d/20-iconv.ini, /etc/php/8.3/apache2/conf.d/20-mysqli.ini, /etc/php/8.3/apache2/conf.d/20-pdo_mysql.ini, /etc/php/8.3/apache2/conf.d/20-phar.ini, /etc/php/8.3/apache2/conf.d/20-posix.ini, /etc/php/8.3/apache2/conf.d/20-readline.ini, /etc/php/8.3/apache2/conf.d/20-shmop.ini, /etc/php/8.3/apache2/conf.d/20-sockets.ini, /etc/php/8.3/apache2/conf.d/20-sysvmsg.ini, /etc/php/8.3/apache2/conf.d/20-sysvsem.ini, /etc/php/8.3/apache2/conf.d/20-sysvshm.ini, /etc/php/8.3/apache2/conf.d/20-tokenizer.ini
PHP API	20230831
PHP Extension	20230831
Zend Extension	420230831
Zend Extension Build	API420230831,NTS
PHP Extension Build	API20230831,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
Zend Max Execution Timers	disabled
IPv6 Support	enabled
DTrace Support	disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2, tlsv1.3
Registered Stream Filters	zlib.*, string.rot13, string.toupper, string.tolower, convert.*, consumed, dechunk, convert.iconv.*

This program makes use of the Zend Scripting Language Engine:
Zend Engine v4.3.6, Copyright (c) Zend Technologies with Zend OPcache v8.3.6, Copyright (c), by Zend Technologies

CREATE MYSQL DATABASE AND USER

Log into MySQL and create a database for WordPress

```
$ sudo mysql

$ CREATE DATABASE wordpress;
```

Create a user with privileges

```
$ CREATE USER 'wordpressuser'@'localhost' IDENTIFIED BY 'password';

$ GRANT ALL PRIVILEGES ON wordpress.* TO 'wordpressuser'@'localhost';

$ FLUSH PRIVILEGES;

$ EXIT;
```

DOWNLOAD AND EXTRACT WORDPRESS

To navigate to the temporary directory, download the latest WordPress version, and move it to the Apache document root, you can use the following commands in a Linux terminal

```
$ cd /tmp

$ wget https://wordpress.org/latest.tar.gz

$ tar xf latest.tar.gz

$ mv wordpress /var/www/html/
```

SET PERMISSIONS : **Adjust file permissions to ensure Apache**

```
$ sudo chown -R www-data:www-data /var/www/html/wordpress $ sudo chmod -R 755 /var/www/html/wordpress
```

CONFIGURE APACHE FOR WORDPRESS

Create a new virtual host configuration file

- **PATH :** sudo nano /etc/apache2/sites-available/wordpress.conf

ADD THE FOLLOWING CONFIGURATION

```
<VirtualHost *:80>
ServerAdmin webmaster@example.com
DocumentRoot /var/www/html/wordpress
ServerName your_domain_or_IP
<Directory /var/www/html/wordpress>
Options FollowSymLinks
AllowOverride All
Require all granted
</Directory>
ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

SAVE AND EXIT

Enable The Site and Restart Apache

$ *sudo a2ensite wordpress*

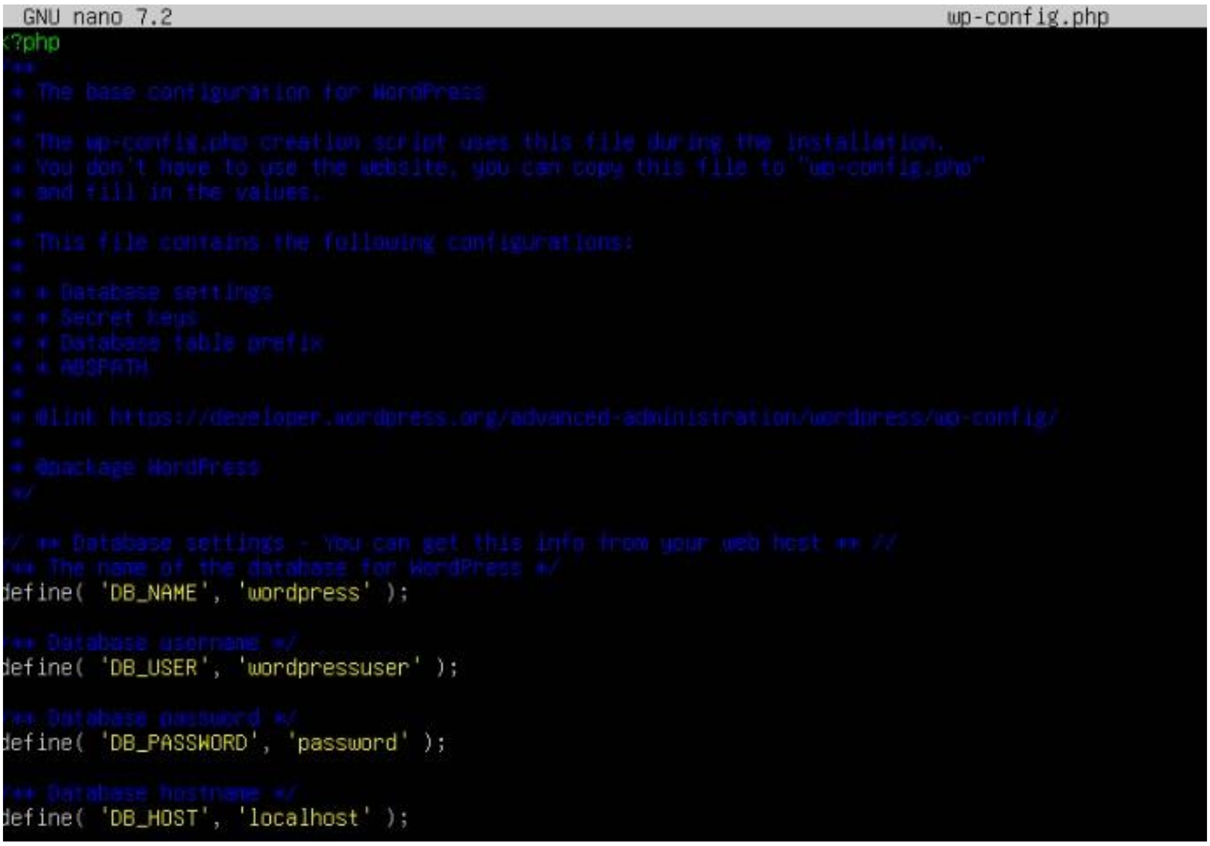
$ *sudo systemctl restart apache2*

SETTING UP WORDPRESS INSTALLATION

$ *cd /var/www/html/wordpress*

$ *sudo mv wp-config-sample.php wp-config.php*

$ *sudo nano wp-config.php*



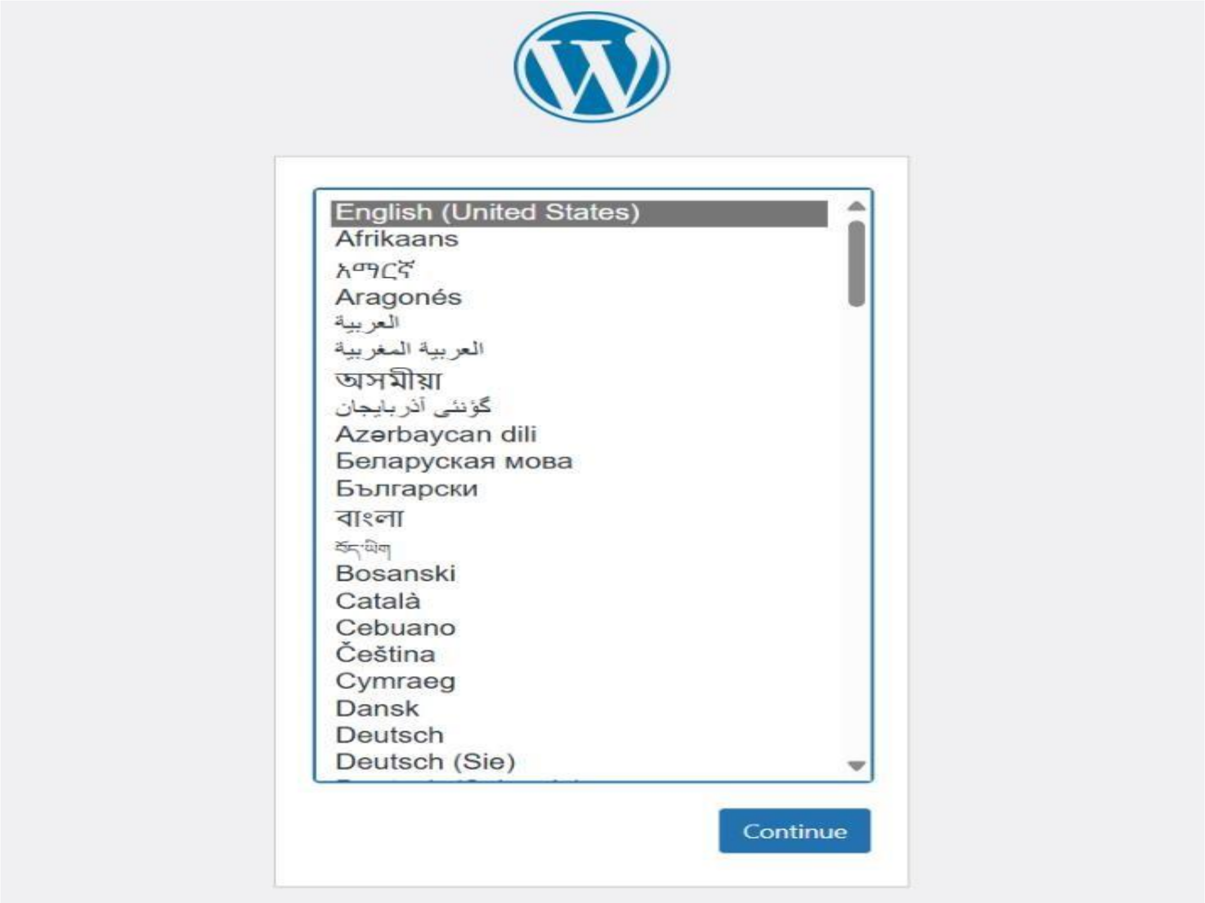
```
  GNU nano 7.2                                                    wp-config.php
<?php
/**
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the installation.
 * You don't have to use the website, you can copy this file to "wp-config.php"
 * and fill in the values.
 *
 * This file contains the following configurations:
 *
 * * Database settings
 * * Secret keys
 * * Database table prefix
 * * ABSPATH
 *
 * @link https://developer.wordpress.org/advanced-administration/wordpress/wp-config/
 *
 * @package WordPress
 */

// ** Database settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'wordpress' );

/** Database username */
define( 'DB_USER', 'wordpressuser' );

/** Database password */
define( 'DB_PASSWORD', 'password' );

/** Database hostname */
define( 'DB_HOST', 'localhost' );
```

Make sure to replace 'wordpress', 'wordpressuser', and 'password' with your actual database name, database user, and password, respectively

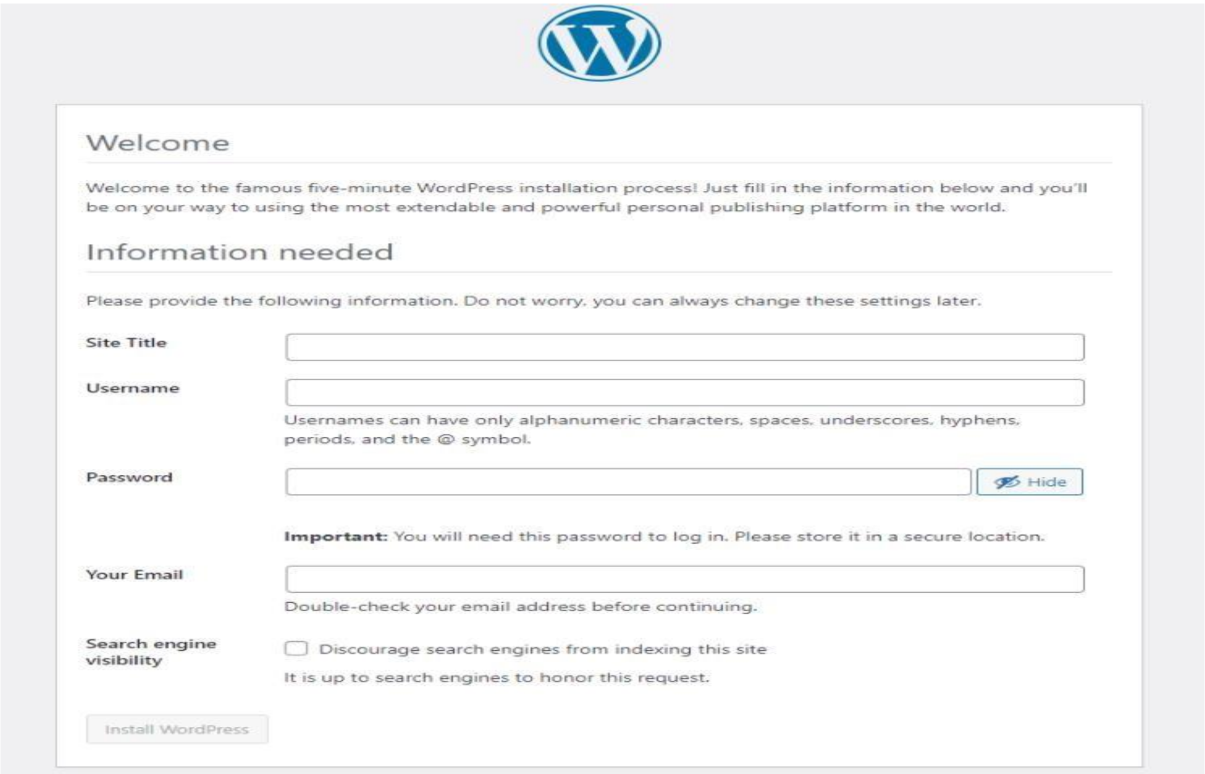
After completing the wordpress configuration file, you need to open to web browser and navigate to the following URL

➔ **http://server-ip/wordpress**

Then it appears the installation phase of the wordpress site, we need to fill up the stages to finish the installation phase.



Select language



Welcome

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. Do not worry, you can always change these settings later.

Site Title

Username

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

Password

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

Your Email

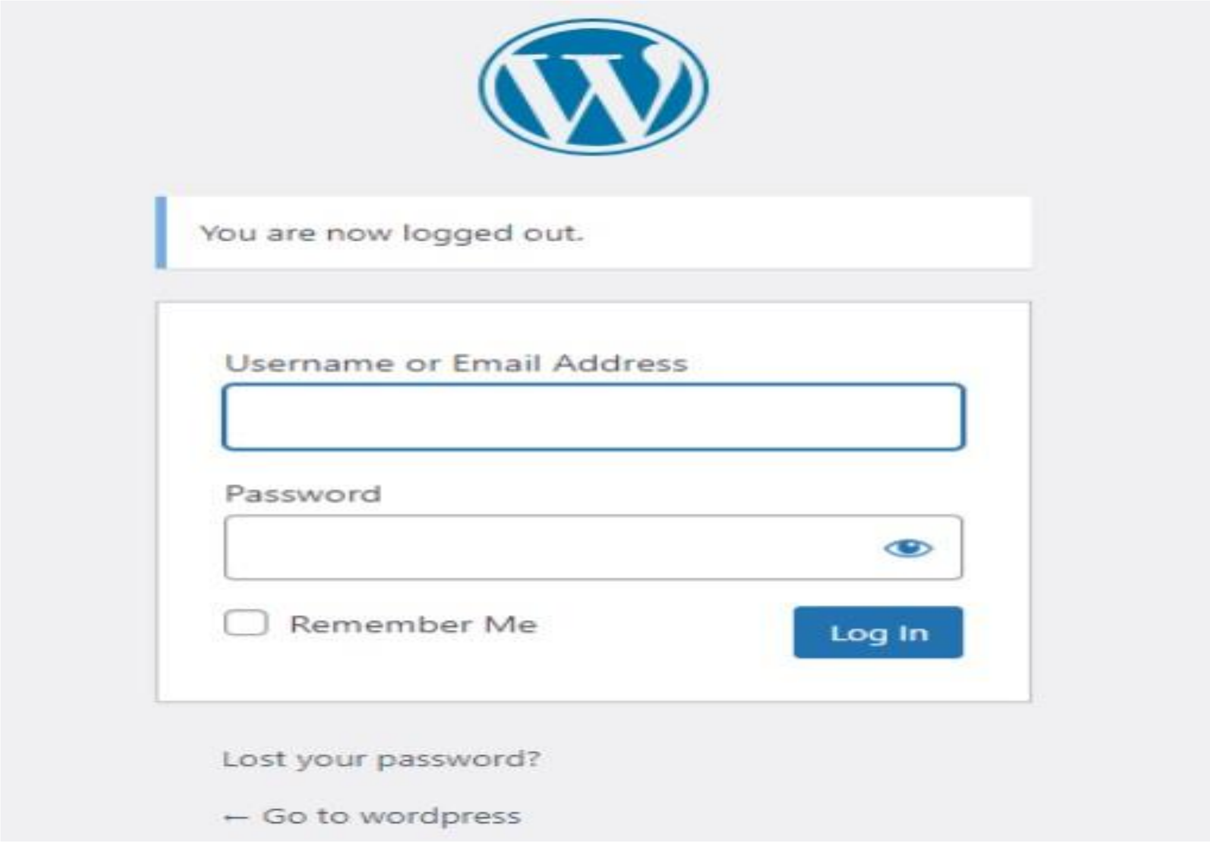
Double-check your email address before continuing.

Search engine visibility

□ Discourage search engines from indexing this site

It is up to search engines to honor this request.

Fill up the information .



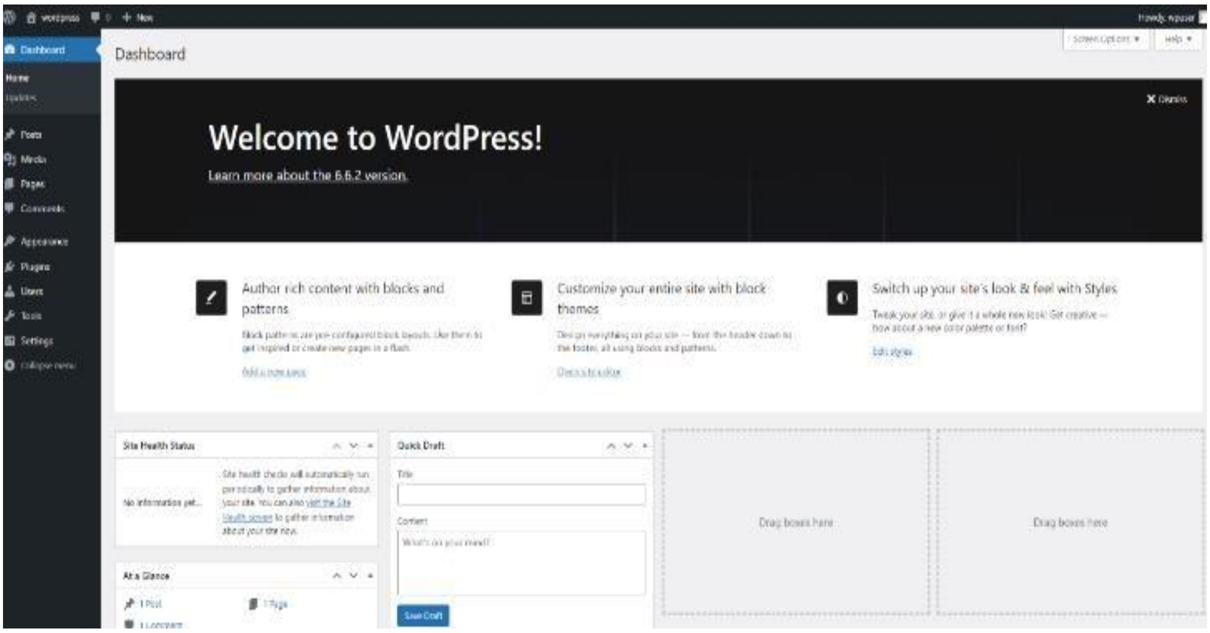
You are now logged out.

Username or Email Address

Password

□ Remember Me

Then Login to Wordpress site using correct username and password



Dashboard

Welcome to WordPress!

Learn more about the 6.6.2 version.

Author rich content with blocks and patterns

Block patterns are pre-configured block layouts. Use them to get inspired or create new pages in a flash.

Customize your entire site with block themes

Design everything on your site — from the header down to the footer, all using blocks and patterns.

Switch up your site's look & feel with Styles

Tweak your site, or give it a whole new look! Get creative — how about a new color palette or font?

Site health checks will automatically run periodically to gather information about your site. You can also visit the Site Health screen to gather information about your site now.

After Successful login it navigate to the wordpress dashboard.

Here we can customise the wordpress site if you are admin by adding users to the wordpress site.

OPTIONAL

CHANGES TO BE MADE IN WORDPRESS CONFIGURATION FILE TO MAKE THE SEVER FULLY DYNAMIC .

```
$currenthost = 'http://'.$_SERVER['SERVER_ADDR'];

$currentpath = preg_replace('@/+$@','',dirname($_SERVER['SCRIPT_NAME']));

$currentpath = preg_replace('/\/wp.+/','',$currentpath);

$siteurl = $currenthost.$currentpath;


define('WP_HOME',$siteurl); define('WP_SITEURL',$siteurl);

define('WP_CONTENT_URL',$siteurl.'/wp-content');

define('WP_PLUGIN_URL',$siteurl.'/wp-content/plugins');

define('DOMAIN_CURRENT_SITE',$siteurl);

@define('ADMIN_COOKIE_PATH', './');
```

Restart apache2 service

```
$ sudo systemctl restart apache2
```

INSTALL AND CONFIGURE SPLUNK UNIVERSAL FORWARDER

Visit Splunk official site (https://www.splunk.com/) OR Get the download link from “copy wget link

OPEN UBUNTU SERVRER

```
$ cd /tmp
```

Download Splunk forwarder

```
$ wget --inet4-only https://download.splunk.com/products/universalforwarder/releases/9.3.2/linux/splunkforwarder-9.3.2-d8bb32809498-Linux-x86_64.tgz
```

Install Splunk forwarder

```
$ sudo tar -xvzf splunkforwarder-9.3.2-d8bb32809498-Linux-x86_64.tgz -C /opt cd
/opt/splunkforwarder/bin
```

Start Splunk and accept the license agreement

```
$ sudo ./splunk start --accept-license
```

Universal Forwarder to start automatically

```
$ sudo ./splunk enable boot-start
```

SETTING UP SPLUNK UNIVERSAL FORWARDER ON UBUNTU SERVER FOR LOG MONITORING

Execute the following commands to modify output configuration file. If the file doesn't exist, create it

```
$ sudo nano /opt/splunkforwarder/etc/system/local/outputs.conf
```

Add the following lines

```
[tcpout] defaultGroup = splunk-group [tcpout:splunk-group] server =
<splunk_server_ip>:<splunk_listener_port>
```

```
$ sudo nano /opt/splunkforwarder/etc/system/local/inputs.conf
```

Add the following lines

```
[monitor:///var/log/apache2/access.log] sourcetype = access_combined index = main
[monitor:///var/log/apache2/error.log
] sourcetype = apache_error index = main
[monitor:///var/log/apache2/other_vhosts_access.log]
sourcetype = apache_error
index = main
```

Restart Splunk forwarder

```
/opt/splunkforwarder/bin/splunk restart
```

ADJUST FIREWALL SETTINGS FOR SPLUNK UNIVERSAL FORWARDER COMMUNICATION

```
$ ufw allow<splunk_listner_port>/tcp
$ ufw allow 80/tcp
$ ufw allow 22/tcp
$ ufw allow 443/tcp
$ ufw reload
```

Restart Splunk Universal Forwarder

```
$ /opt/splunkforwarder/bin/splunk restart
```

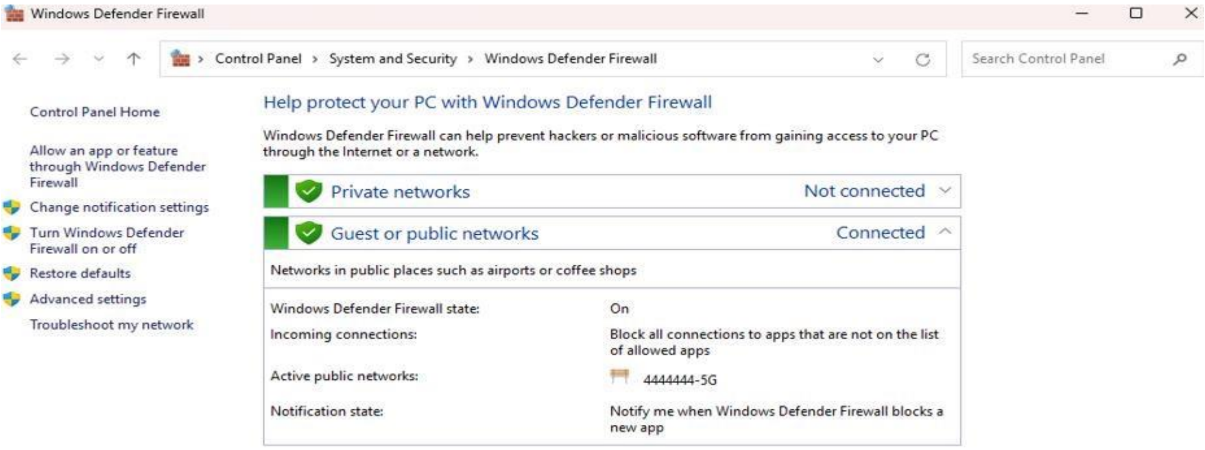
(These steps will help to ensure proper log monitoring and firewall settings for effective communication with splunk server.)

PREPARING SPLUNK SERVER AND CONNECTING IT TO SPLUNK ENTERPRISE FOR WORDPRESS LOG TRACKING

Configuring Windows Firewall for Splunk Universal Forwarder:

Access Windows Firewall Settings:

Navigate to Windows settings and select Windows Defender Firewall.


Windows Defender Firewall
Control Panel › System and Security › Windows Defender Firewall
Search Control Panel
Control Panel Home
Allow an app or feature through Windows Defender Firewall
Change notification settings
Turn Windows Defender Firewall on or off
Restore defaults
Advanced settings
Troubleshoot my network

Help protect your PC with Windows Defender Firewall

Windows Defender Firewall can help prevent hackers or malicious software from gaining access to your PC through the Internet or a network.

Private networks — Not connected

Guest or public networks — Connected

Networks in public places such as airports or coffee shops

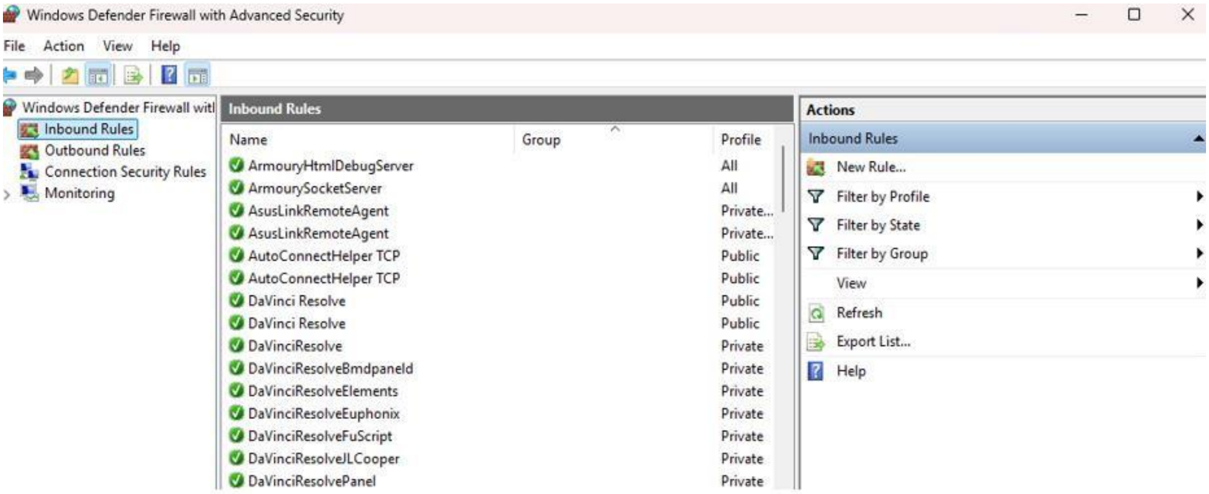
Windows Defender Firewall state: On

Incoming connections: Block all connections to apps that are not on the list of allowed apps

Active public networks: 4444444-5G

Notification state: Notify me when Windows Defender Firewall blocks a new app

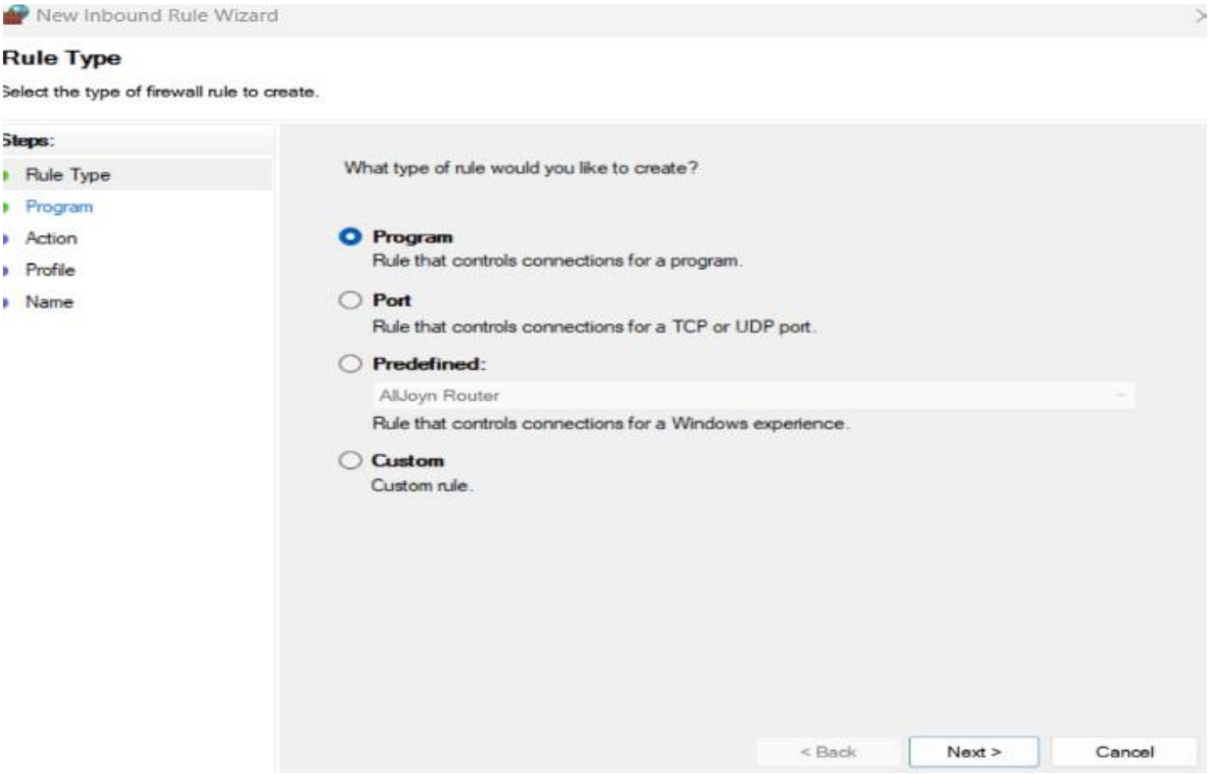
"Advanced settings"



Inbound Rules

Name	Group	Profile
ArmouryHtmlDebugServer		All
ArmourySocketServer		All
AsusLinkRemoteAgent		Private...
AsusLinkRemoteAgent		Private...
AutoConnectHelper TCP		Public
AutoConnectHelper TCP		Public
DaVinci Resolve		Public
DaVinci Resolve		Public
DaVinciResolve		Private
DaVinciResolveBmdpaneld		Private
DaVinciResolveElements		Private
DaVinciResolveEuphonix		Private
DaVinciResolveFuScript		Private
DaVinciResolveJLCooper		Private
DaVinciResolvePanel		Private

Click on the **Inbound Rules** and select **new rules** from the actions



Rule Type

Select the type of firewall rule to create.

Steps:

- Rule Type
- Program
- Action
- Profile
- Name

What type of rule would you like to create?

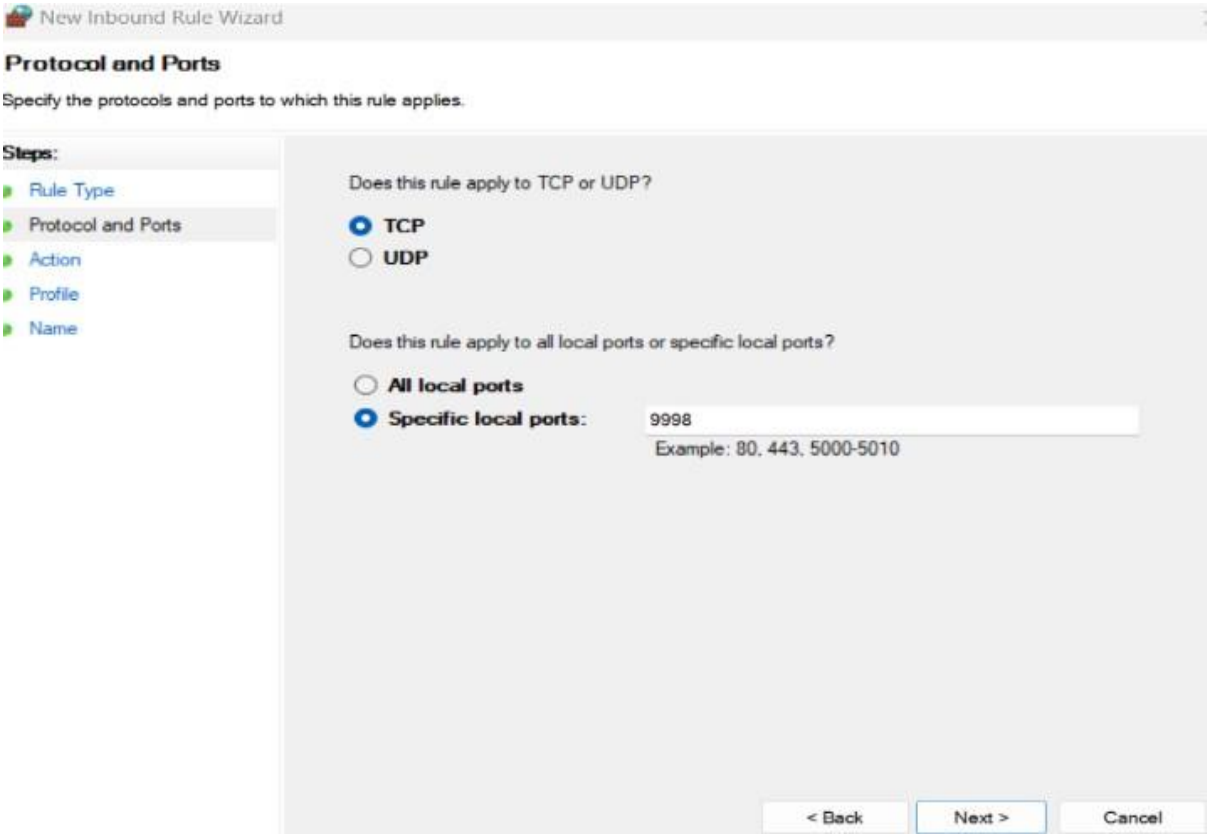
● Program
Rule that controls connections for a program.

○ Port
Rule that controls connections for a TCP or UDP port.

○ Predefined:
AllJoyn Router
Rule that controls connections for a Windows experience.

○ Custom
Custom rule.

Select **port** and click **next**



Protocol and Ports

Specify the protocols and ports to which this rule applies.

Steps:

- Rule Type
- Protocol and Ports
- Action
- Profile
- Name

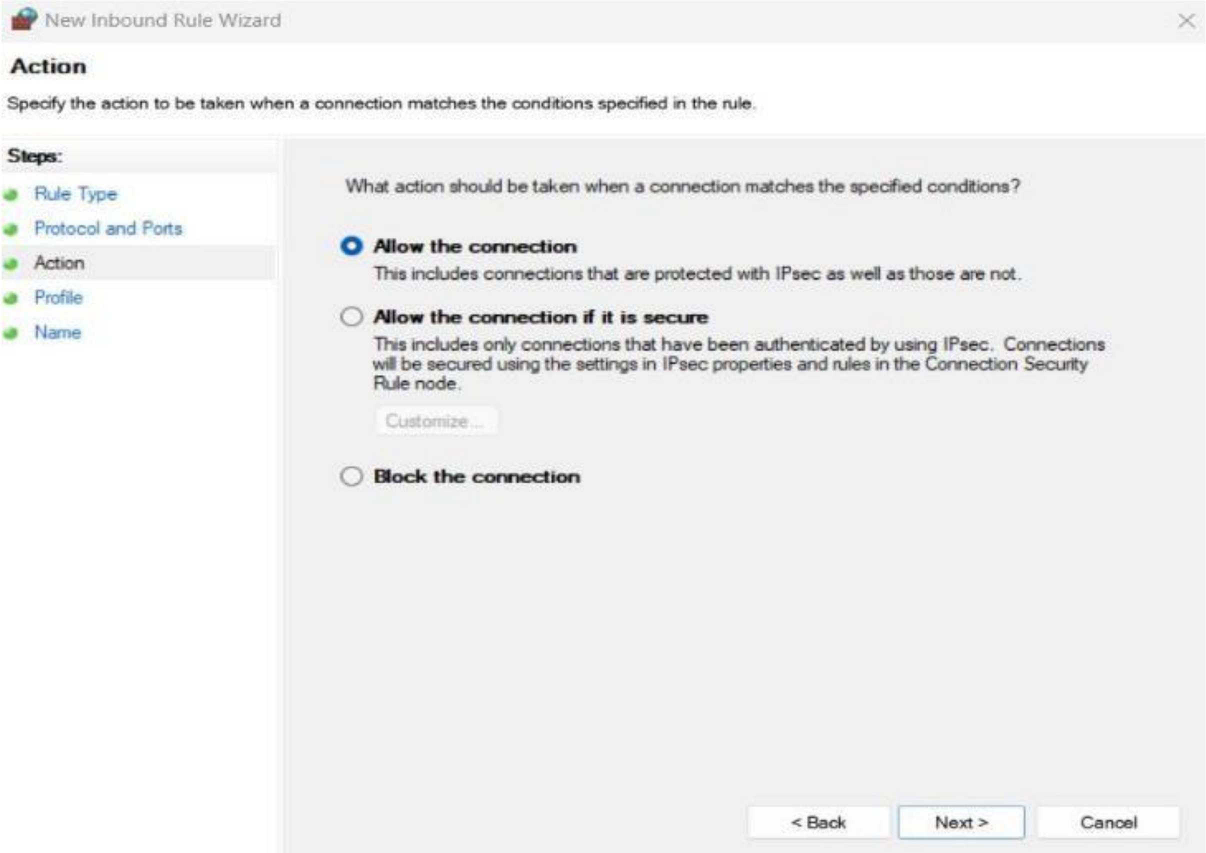
Does this rule apply to TCP or UDP?

● TCP
○ UDP

Does this rule apply to all local ports or specific local ports?

○ All local ports
● Specific local ports: 9998
Example: 80, 443, 5000-5010

Select **TCP** and select **specified local port(eg:9998)** and click **Next.**



Action

Specify the action to be taken when a connection matches the conditions specified in the rule.

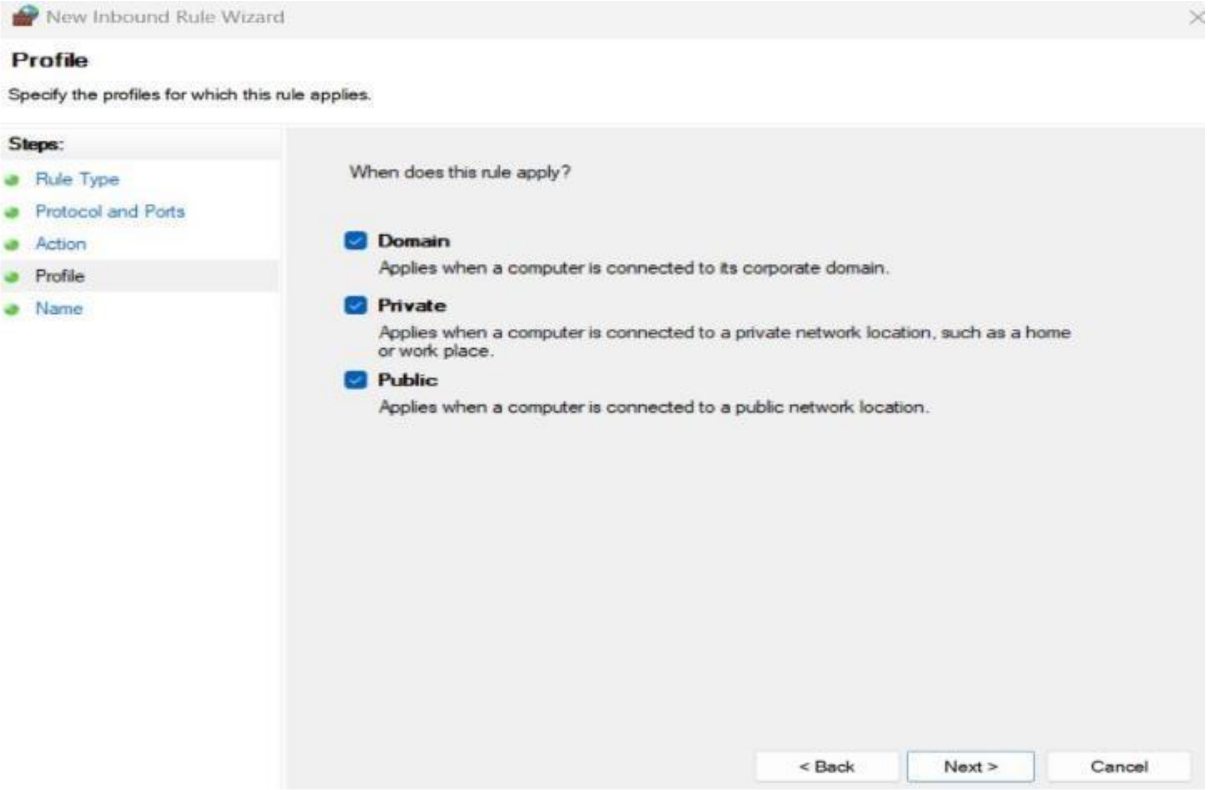
What action should be taken when a connection matches the specified conditions?

● **Allow the connection**
This includes connections that are protected with IPsec as well as those are not.

○ **Allow the connection if it is secure**
This includes only connections that have been authenticated by using IPsec. Connections will be secured using the settings in IPsec properties and rules in the Connection Security Rule node.

○ **Block the connection**

Select **Allow the connection** and click on **Next.**



Profile

Specify the profiles for which this rule applies.

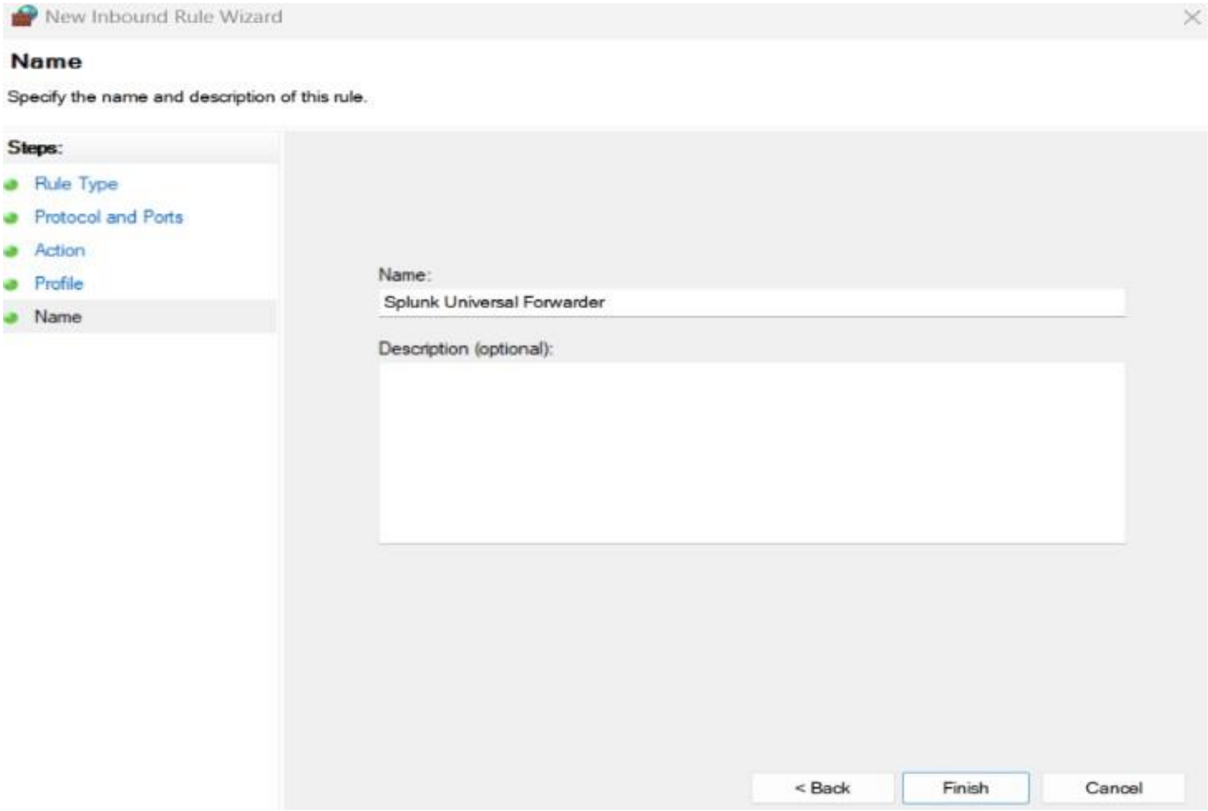
When does this rule apply?

☑ **Domain**
Applies when a computer is connected to its corporate domain.

☑ **Private**
Applies when a computer is connected to a private network location, such as a home or work place.

☑ **Public**
Applies when a computer is connected to a public network location.

Select **Domain, Private and Public** and click on **Next.**



New Inbound Rule Wizard

Name

Specify the name and description of this rule.

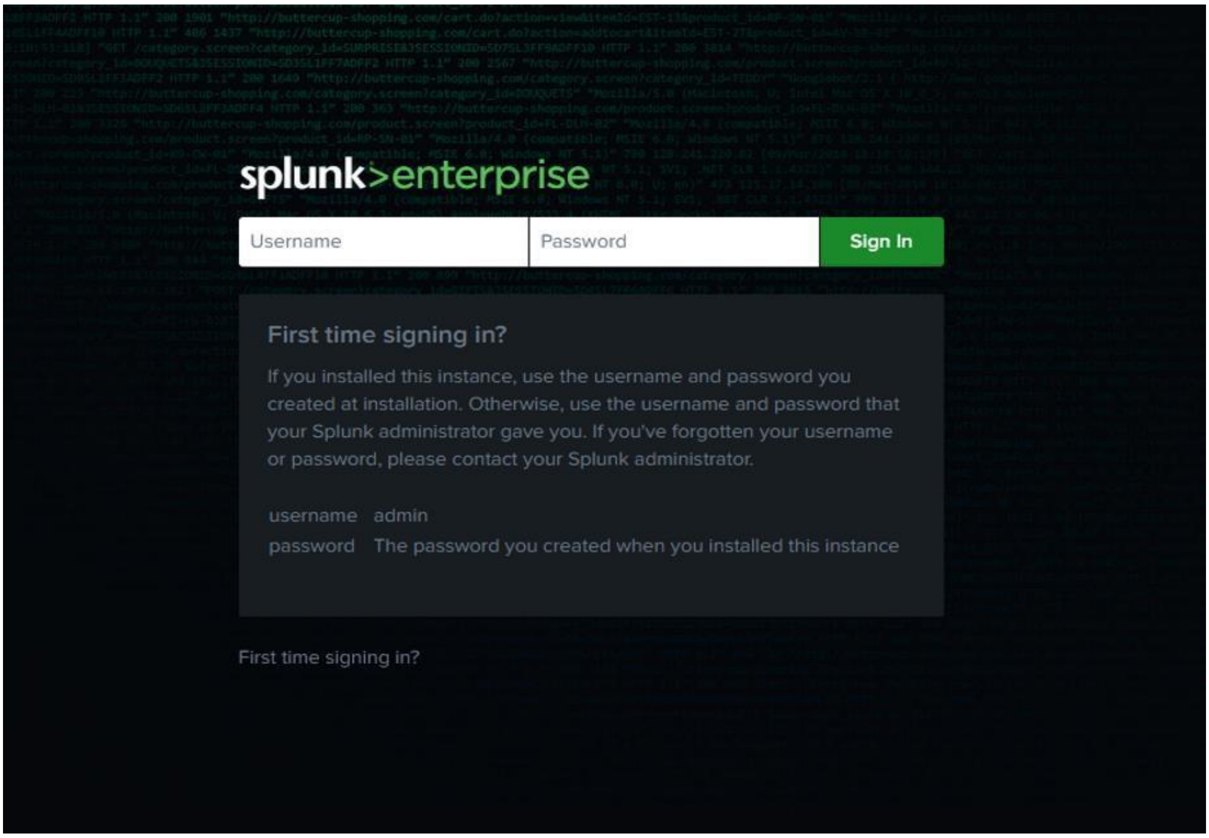
Steps:
- Rule Type
- Protocol and Ports
- Action
- Profile
- Name

Name: Splunk Universal Forwarder

Description (optional):

Name the Inbound rule and click on **Finish.**

LOGIN TO THE SPLUNK ENTERPRISE

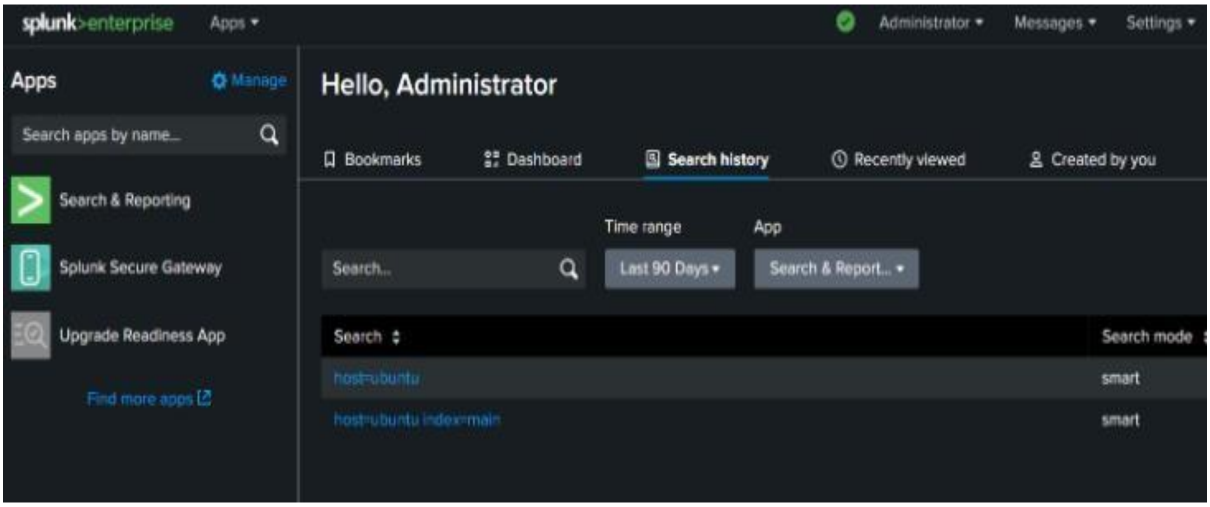


First time signing in?

If you installed this instance, use the username and password you created at installation. Otherwise, use the username and password that your Splunk administrator gave you. If you've forgotten your username or password, please contact your Splunk administrator.

username admin
password The password you created when you installed this instance

First time signing in?



Search	Search mode
host=ubuntu	smart
host=ubuntu index=main	smart

Go to **setting**

Click on **Forwarding and Receiving**

Click on **Configure Receiving**

Click on the **New receiving Port**

Add the listening **port (eg:9998)** and save it

Open splunk forwarder and run this command

$ *sudo su*

/opt/splunkforwarder/bin/splunk add monitor /var/log/syslog -index main -sourcetype syslog

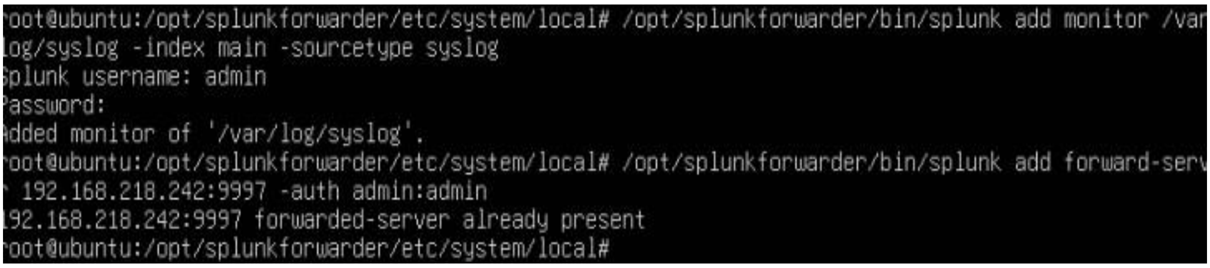
/opt/splunkforwarder/bin/splunk add forward-server your_splunk_server_ip:port -auth admin:username_of_your_splunkforwarder

Replace the username_of_your_splunk forwarder with your splunk forwarder username

$ */opt/splunkforwarder/bin/splunk restart*

Now check if the forwarding is active or not

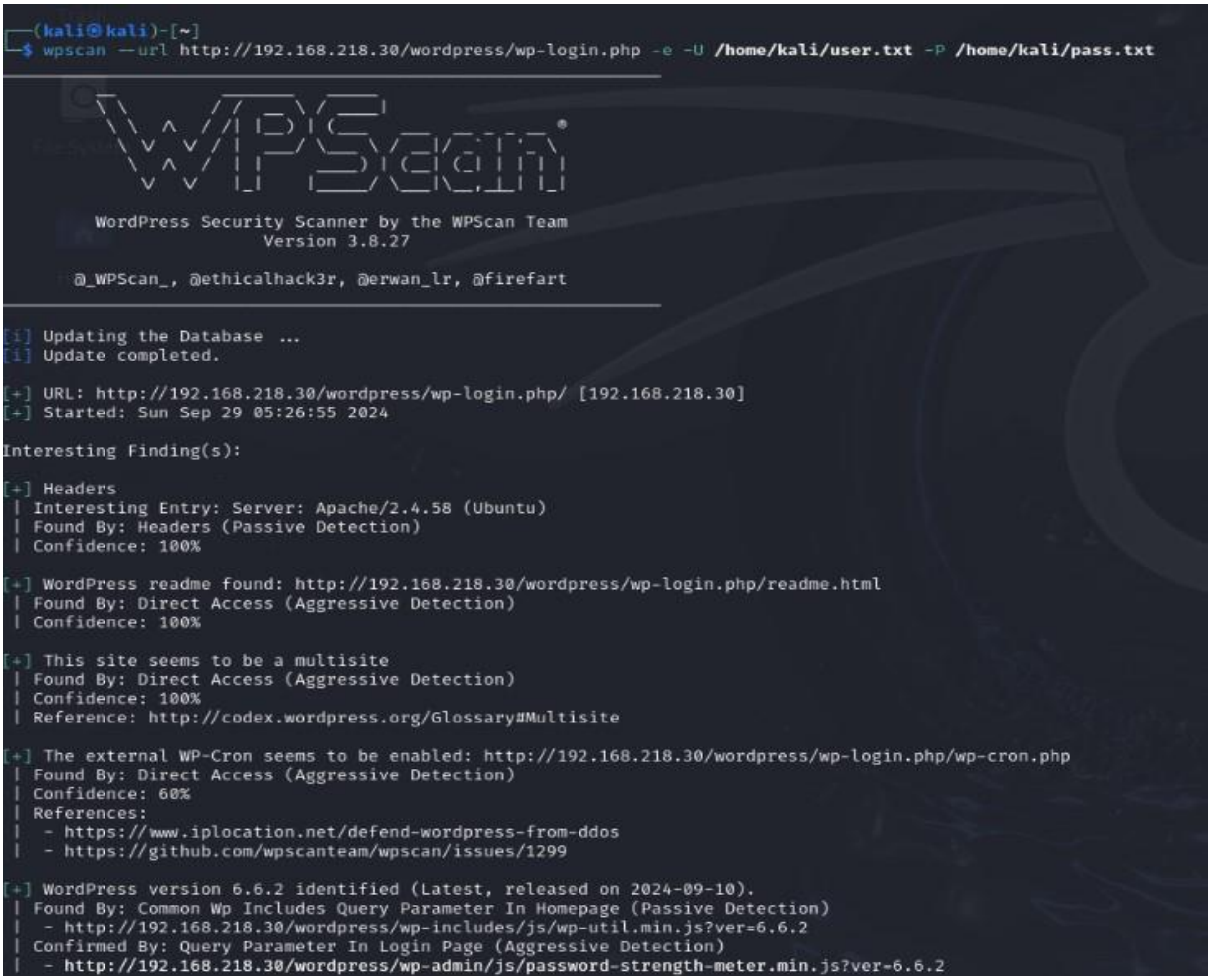
$ */opt/splunkforwarder/bin/splunk list forward-server*



```
oot@ubuntu:/opt/splunkforwarder/etc/system/local# /opt/splunkforwarder/bin/splunk add monitor /var
log/syslog -index main -sourcetype syslog
plunk username: admin
assword:
dded monitor of '/var/log/syslog'.
oot@ubuntu:/opt/splunkforwarder/etc/system/local# /opt/splunkforwarder/bin/splunk add forward-serv
 192.168.218.242:9997 -auth admin:admin
92.168.218.242:9997 forwarded-server already present
oot@ubuntu:/opt/splunkforwarder/etc/system/local#
```

BRUTEFORCING THE WORDPRESS LOGIN PAGE USING WPSCAN AND CAPTURE ITS LOG USING SPLUNK

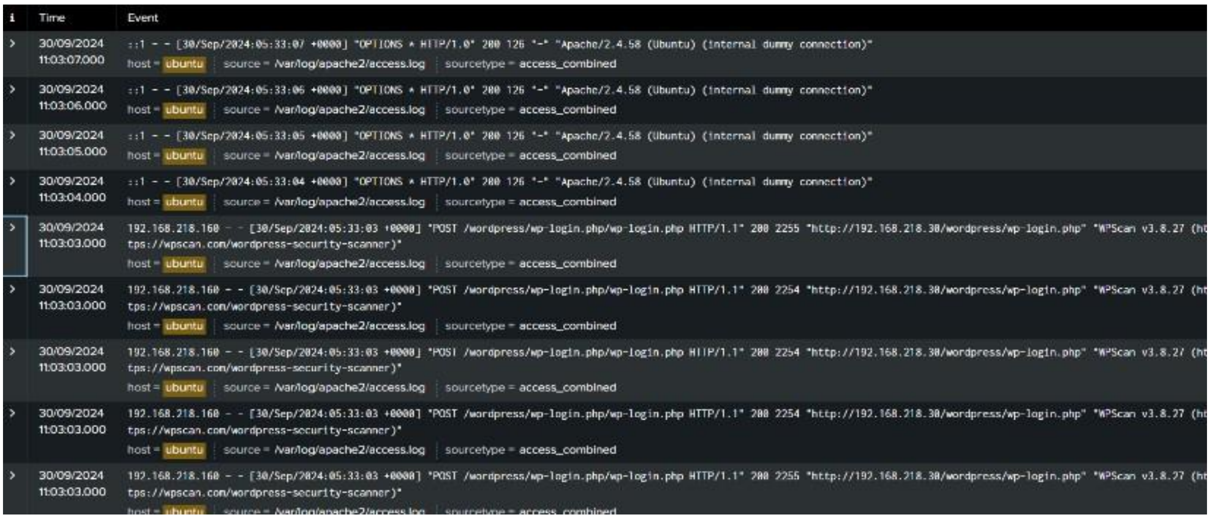
Use Kali Linux Command line Terminal and Perform the WPScan for the wordpress login page



```
┌──(kali㉿kali)-[~]
└─$ wpscan --url http://192.168.218.30/wordpress/wp-login.php -e -U /home/kali/user.txt -P /home/kali/pass.txt
_______________________________________________________________

         WordPress Security Scanner by the WPScan Team
                         Version 3.8.27

       @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________

[i] Updating the Database ...
[i] Update completed.

[+] URL: http://192.168.218.30/wordpress/wp-login.php/ [192.168.218.30]
[+] Started: Sun Sep 29 05:26:55 2024

Interesting Finding(s):

[+] Headers
 | Interesting Entry: Server: Apache/2.4.58 (Ubuntu)
 | Found By: Headers (Passive Detection)
 | Confidence: 100%

[+] WordPress readme found: http://192.168.218.30/wordpress/wp-login.php/readme.html
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%

[+] This site seems to be a multisite
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%
 | Reference: http://codex.wordpress.org/Glossary#Multisite

[+] The external WP-Cron seems to be enabled: http://192.168.218.30/wordpress/wp-login.php/wp-cron.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 60%
 | References:
 |  - https://www.iplocation.net/defend-wordpress-from-ddos
 |  - https://github.com/wpscanteam/wpscan/issues/1299

[+] WordPress version 6.6.2 identified (Latest, released on 2024-09-10).
 | Found By: Common Wp Includes Query Parameter In Homepage (Passive Detection)
 |  - http://192.168.218.30/wordpress/wp-includes/js/wp-util.min.js?ver=6.6.2
 | Confirmed By: Query Parameter In Login Page (Aggressive Detection)
 |  - http://192.168.218.30/wordpress/wp-admin/js/password-strength-meter.min.js?ver=6.6.2
```

By doing this we get the username and password of the wordpress admin

Open the Splunk enterprise and search for “host=hostname index=main”



```
i  Time         Event
>  30/09/2024   ::1 - - [30/Sep/2024:05:33:07 +0000] "OPTIONS * HTTP/1.0" 200 126 "-" "Apache/2.4.58 (Ubuntu) (internal dummy connection)"
   11:03:07.000 host = ubuntu | source = /var/log/apache2/access.log | sourcetype = access_combined
>  30/09/2024   ::1 - - [30/Sep/2024:05:33:06 +0000] "OPTIONS * HTTP/1.0" 200 126 "-" "Apache/2.4.58 (Ubuntu) (internal dummy connection)"
   11:03:06.000 host = ubuntu | source = /var/log/apache2/access.log | sourcetype = access_combined
>  30/09/2024   ::1 - - [30/Sep/2024:05:33:05 +0000] "OPTIONS * HTTP/1.0" 200 126 "-" "Apache/2.4.58 (Ubuntu) (internal dummy connection)"
   11:03:05.000 host = ubuntu | source = /var/log/apache2/access.log | sourcetype = access_combined
>  30/09/2024   ::1 - - [30/Sep/2024:05:33:04 +0000] "OPTIONS * HTTP/1.0" 200 126 "-" "Apache/2.4.58 (Ubuntu) (internal dummy connection)"
   11:03:04.000 host = ubuntu | source = /var/log/apache2/access.log | sourcetype = access_combined
>  30/09/2024   192.168.218.160 - - [30/Sep/2024:05:33:03 +0000] "POST /wordpress/wp-login.php/wp-login.php HTTP/1.1" 200 2255 "http://192.168.218.30/wordpress/wp-login.php" "WPScan v3.8.27 (ht
   11:03:03.000 tps://wpscan.com/wordpress-security-scanner)"
                host = ubuntu | source = /var/log/apache2/access.log | sourcetype = access_combined
>  30/09/2024   192.168.218.160 - - [30/Sep/2024:05:33:03 +0000] "POST /wordpress/wp-login.php/wp-login.php HTTP/1.1" 200 2254 "http://192.168.218.30/wordpress/wp-login.php" "WPScan v3.8.27 (ht
   11:03:03.000 tps://wpscan.com/wordpress-security-scanner)"
                host = ubuntu | source = /var/log/apache2/access.log | sourcetype = access_combined
>  30/09/2024   192.168.218.160 - - [30/Sep/2024:05:33:03 +0000] "POST /wordpress/wp-login.php/wp-login.php HTTP/1.1" 200 2254 "http://192.168.218.30/wordpress/wp-login.php" "WPScan v3.8.27 (ht
   11:03:03.000 tps://wpscan.com/wordpress-security-scanner)"
                host = ubuntu | source = /var/log/apache2/access.log | sourcetype = access_combined
>  30/09/2024   192.168.218.160 - - [30/Sep/2024:05:33:03 +0000] "POST /wordpress/wp-login.php/wp-login.php HTTP/1.1" 200 2254 "http://192.168.218.30/wordpress/wp-login.php" "WPScan v3.8.27 (ht
   11:03:03.000 tps://wpscan.com/wordpress-security-scanner)"
                host = ubuntu | source = /var/log/apache2/access.log | sourcetype = access_combined
>  30/09/2024   192.168.218.160 - - [30/Sep/2024:05:33:03 +0000] "POST /wordpress/wp-login.php/wp-login.php HTTP/1.1" 200 2255 "http://192.168.218.30/wordpress/wp-login.php" "WPScan v3.8.27 (ht
   11:03:03.000 tps://wpscan.com/wordpress-security-scanner)"
                host = ubuntu | source = /var/log/apache2/access.log | sourcetype = access_combined
```

CONCLUSION

By successfully setting up a server and installing WordPress, this project provides valuable hands-on experience in server management and web application deployment. Configuring WordPress and exploiting its vulnerabilities not only demonstrates practical knowledge of security gaps but also highlights the importance of understanding common attack vectors like brute force attacks. These exercises deepen technical expertise and enhance the ability to identify and mitigate risks associated with web applications.

Furthermore, the integration of Splunk for log monitoring adds a critical layer of analysis and oversight. By analyzing logs, you gain insight into application behavior, detect potential threats, and ensure the overall security of the system. This comprehensive approach equips you with skills essential for realworld scenarios, emphasizing the importance of maintaining, securing, and monitoring web applications to deliver optimal performance and protection against evolving threats.