Proof of Concept Technical Solution for the Marconi Law Firm, LLC. (WordPress Website)

Orlando, Florida

ELIJAH MONTGOMERY

MONTHEED MARTIN

Table of Contents

Inventory

EQUIPMENT	OPERATIING SYSTEM	ADDITIONAL INFO	IP ADDRESS
Router/Custom Network	- (Firewall VM)	-Firewall VM	10.10.229.1
Docker	Rocky 8 (-Docker)	Ghost Container	10.10.229.11
NginX Reverse Proxy	Rocky 8 (-Nginx)	Reverse Proxy	10.10.229.10
WordPress	Ubuntu	LAMP Stack running WordPress	10.10.229.12

Custom Network

NETWORK NAME	SUBNET IP	SUBNET MASK	DNS	GATEWAY
ITE229	10.10.229.0	255.255.255.0	10.10.229.1	10.10.229.1

IDs and Passwords

ACCOUNT	USER ID	PASSWORD
Rocky8-Docker Root User	root	Fullsail1!
Rocky8-Nginx Root User	root	Fullsail1!
Ubuntu Root User	Root	Fullsail1!
MySQL Root User	root@localhost	[randompassword]
MySQL WordPress User	WordPressUser	[randompassword]
WordPress Admin	admin	[randompassword]

Preface

This document will serve as proof of concept to Mr. Marconi for creating his WordPress website for his law firm and as audit documentation.

The purpose of audit documentation is to provide a comprehensive record of the organization's information technology infrastructure and security controls and processes. It plays a crucial role in providing transparency, accountability, and QA/QC regarding an organization's cybersecurity controls and practices. It enables organizations to demonstrate compliance, identify areas for improvement, and make informed decisions to strengthen their overall organizational cybersecurity.

Audit documentation serves several important purposes:

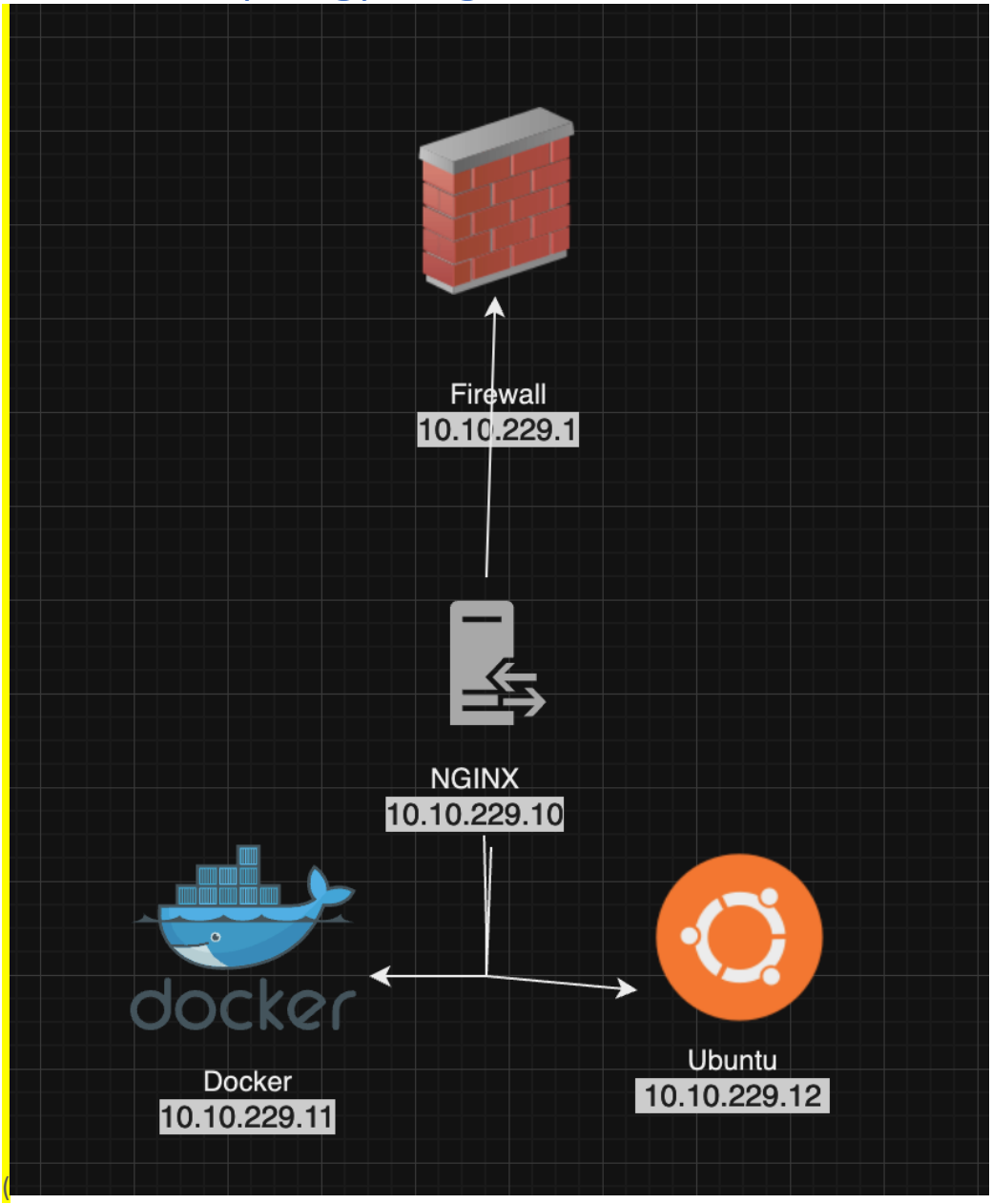
- Compliance: Evidence that an organization has undergone a thorough examination of its systems. It helps validate that the organization has implemented appropriate controls to protect its information systems and sensitive data.

- Validation: Verification of the effectiveness and adequacy of cybersecurity controls. It provides detailed information about the design, implementation, and operation of these controls, enabling reviewers to assess their reliability and identify any gaps or weaknesses.

- Records Maintenance: Historical record of cybersecurity audits conducted over time. It enables organizations to track their progress, identify trends, and evaluate the effectiveness actions taken. It also serves as reference for future audits and allows auditors to understand the current cybersecurity implemented and facilitates a more targeted approach to future cybersecurity updates and audits.

- Decision-making Support: Valuable insights and information that can support decision-making processes. It allows management to make informed decisions about allocating resources, prioritizing cybersecurity investments, and addressing identified risks and vulnerabilities.

Network Topology Diagram



SSH Install & Access

OpenSSH Install (Ubuntu)

Legible, annotated screenshots AND written instructions/commands required

```
root@UbuntuElijah:~# apt install openssh-client -y
```

Node.js Application (Ghost) on Docker

Update Rocky8-Docker Host Name

Legible, annotated screenshots AND written instructions/commands required

```
[root@DockerMontgomery ~]#
```

Update Rocky8-Docker

Legible, annotated screenshots AND written instructions/commands required

```
[root@DockerMontgomery ~]# yum update -y
```

Install EPEL Packages

Legible, annotated screenshots AND written instructions/commands required

```
[root@DockerMontgomery ~]# yum install epel-release -y
```

Install Nano Editor

Legible, annotated screenshots AND written instructions/commands required

```
[root@DockerMontgomery ~]# yum install nano -y
```

Docker CE

Set up stable repository

Legible, annotated screenshots AND written instructions/commands required

This section not shown in video. Therefore, please research how to set up a stable repository for Docker.

```
[root@DockerMontgomery ~]# yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo
```

Install Docker CE

Legible, annotated screenshots AND written instructions/commands required

```
[root@DockerMontgomery ~]# yum install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin -y
```

Verify Docker version

Legible, annotated screenshots AND written instructions/commands required

```
[root@DockerMontgomery ~]# docker -v
Docker version 26.1.3, build b72abbb
```

Initialize Docker

Start Docker

Legible, annotated screenshots AND written instructions/commands required

```
[root@DockerMontgomery ~]# systemctl start docker
```

Enable Docker

Legible, annotated screenshots AND written instructions/commands required

```
[root@DockerMontgomery ~]# systemctl enable docker
```

Test Docker (hello-world)

Legible, annotated screenshots AND written instructions/commands required

```
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/

[root@DockerMontgomery ~]#
```

Disable SELinux (/etc/selinux.config)

Legible, annotated screenshots AND written instructions/commands required

```
1
2 # This file controls the state of SELinux on the system.
3 # SELINUX= can take one of these three values:
4 #     enforcing - SELinux security policy is enforced.
5 #     permissive - SELinux prints warnings instead of enforcing.
6 #     disabled - No SELinux policy is loaded.
7 SELINUX=disabled
8 # SELINUXTYPE= can take one of these three values:
9 #     targeted - Targeted processes are protected,
10 #     minimum - Modification of targeted policy. Only selected processes are protected.
11 #     mls - Multi Level Security protection.
12 SELINUXTYPE=targeted
```

Reboot VM

Legible, annotated screenshots AND written instructions/commands required

```
[root@DockerMontgomery ~]# reboot now
```

Confirm SELinux Status

Legible, annotated screenshots AND written instructions/commands required

```
[root@DockerMontgomery ~]# sestatus
SELinux status:                 disabled
```

Install Ghost Docker Container

Legible, annotated screenshots AND written instructions/commands required

```
[root@DockerMontgomery ~]# docker run -d --name ghostttt -p 3001:2368 -e url=http://10.10.229.11:3001 ghost
```

Check for Ghost Container (docker ps)

Ghost Container ID

Legible, annotated screenshots AND written instructions/commands required

```
[root@DockerMontgomery ~]# docker ps -a
CONTAINER ID   IMAGE         COMMAND                  CREATED          STATUS                      PORTS     NAMES
1f8318d23247   ghost         "docker-entrypoint.s…"   54 seconds ago   Exited (2) 45 seconds ago             ghostttt
a878d2595f78   hello-world   "/hello"                 13 minutes ago   Exited (0) 13 minutes ago             ecstatic_gagarin
```

NginX Reverse Proxy

Update Rocky8-Nginx Host Name

Legible, annotated screenshots AND written instructions/commands required

```
[root@NginxMontgomery ~]# nmtui
```

Update Rocky8-Nginx

Legible, annotated screenshots AND written instructions/commands required

```
[root@NginxMontgomery ~]# yum update -y
```

Install EPEL Packages

Legible, annotated screenshots AND written instructions/comm

```
[root@NginxMontgomery ~]# yum install epel-release -y
```

ands required

```
[root@NginxMontgomery ~]# yum install epel-release
```

Install Nano Editor

Legible, annotated screenshots AND written instructions/commands required

```
[root@NginxMontgomery ~]# yum install nano -y
```

Disable SELinux

Legible, annotated screenshots AND written instructions/commands required

```
root@NginxMontgomery:~
  GNU nano 2.9.8                                          /etc/se


# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy is loaded.
SELINUX=disabled
```

Reboot VM

Legible, annotated screenshots AND written instructions/commands required

```
[root@NginxMontgomery ~]# reboot
```

Confirm SELinux Status

Legible, annotated screenshots AND written instructions/commands required

```
user@UbuntuElijah:~$ sestatus

Command 'sestatus' not found, but can be installed with:
```

Rocky Firewall

Stop Firewall

Legible, annotated screenshots AND written instructions/commands required

```
[root@NginxMontgomery ~]# systemctl stop firewalld
```

Disable Firewall

Legible, annotated screenshots AND written instructions/commands required

```
[root@NginxMontgomery ~]# systemctl disable firewalld
Removed /etc/systemd/system/multi-user.target.wants/firewalld.service.
Removed /etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service.
```

NginX

Install NginX

Legible, annotated screenshots AND written instructions/commands required

```
[root@NginxMontgomery ~]# yum install nginx -y
```

Start NginX

Legible, annotated screenshots AND written instructions/commands required

```
[root@NginxMontgomery ~]# systemctl start nginx
```

Enable NginX

Legible, annotated screenshots AND written instructions/commands required

```
[root@NginxMontgomery ~]# systemctl enable nginx
Created symlink /etc/systemd/system/multi-user.target.wants/nginx.service → /usr/lib/systemd/system/nginx.service.
```

Confirm NginX Status

Legible, annotated screenshots AND written instructions/commands required

```
[root@NginxMontgomery ~]# systemctl status nginx
● nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; vendor preset: disabled)
   Active: active (running) since Tue 2024-09-10 13:55:51 EDT; 49s ago
 Main PID: 3973 (nginx)
    Tasks: 3 (limit: 11148)
   Memory: 8.1M
   CGroup: /system.slice/nginx.service
           ├─3973 nginx: master process /usr/sbin/nginx
           ├─3974 nginx: worker process
           └─3975 nginx: worker process

Sep 10 13:55:50 NginxMontgomery systemd[1]: Starting The nginx HTTP and reverse proxy server...
Sep 10 13:55:51 NginxMontgomery nginx[3970]: nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
Sep 10 13:55:51 NginxMontgomery nginx[3970]: nginx: configuration file /etc/nginx/nginx.conf test is successful
Sep 10 13:55:51 NginxMontgomery systemd[1]: Started The nginx HTTP and reverse proxy server.
```

Reverse Proxy for Ghost Site

Edit NginX configuration file

Legible, annotated screenshots AND written instructions/commands required

```
[root@NginxMontgomery ~]# nano /etc/nginx/nginx.conf
        location /blog {
          proxy_pass                          https://10.10.229.11:3001;
          proxy_set_header      Host          $http_host;
          proxy_set_header      X-Real-IP     $remote_addr;
          proxy_set_header      X-Forwarded-For $proxy_add_x_forwarded_for;
          proxy_read_timeout                  900;
        }
```

Reload NginX service

Legible, annotated screen

```
[root@NginxMontgomery ~]# systemctl reload nginx
```

shots AND written instructions/commands required

Delete First Ghost Container

Legible, annotated screenshots AND written instructions required

```
[root@DockerMontgomery ~]# docker rm 1f8318d23247
1f8318d23247
```

Create New Ghost Container

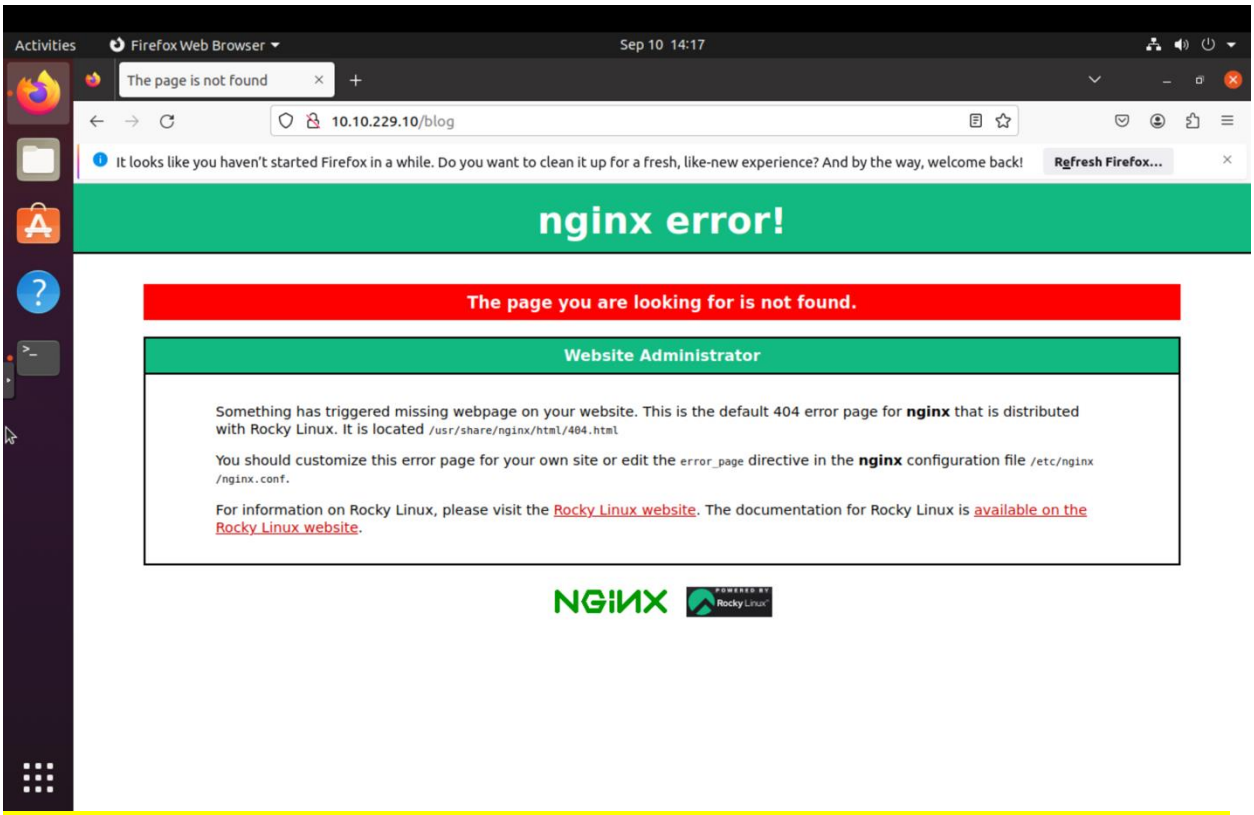
Legible, annotated screenshots AND written instructions required

```
[root@DockerMontgomery ~]# docker run -d --name ghost -p 3001:2368 -e url=http://10.10.229.10/blog ghost
```

Browse to Ghost from Firefox on Ubuntu (10.10.229.10/blog)

Legible, annotated screenshots AND written instructions required

When you browse to Ghost, if you get a web page that says "NginX Error!", don't freak. Just take a screenshot and place it here.



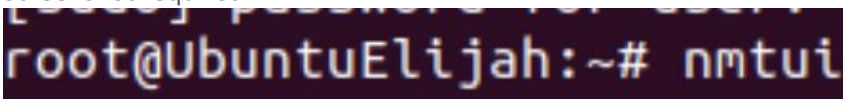
Please remember to complete Appendicies A and B at the end of this document for milestone 1 after you have completed the above steps. Use the "tail" command on your reverse proxy VM. You may need to research this command.

END OF MILESTONE 1

WordPress on Ubuntu - LAMP Stack

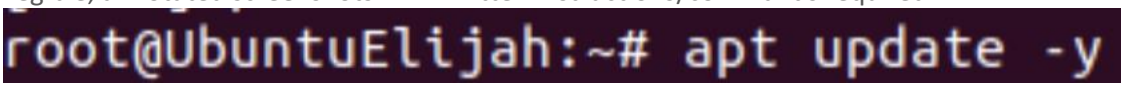
Update Ubuntu Host Name

Screenshot required



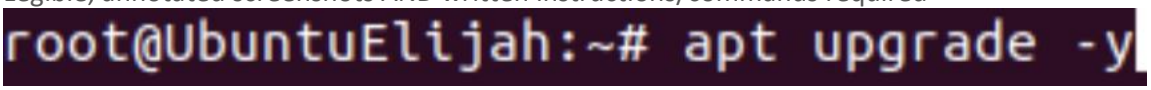
Update Ubuntu

Legible, annotated screenshots AND written instructions/commands required



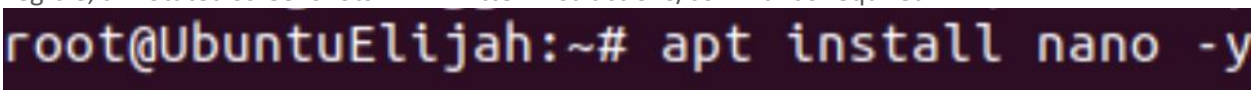
Upgrade Ubuntu

Legible, annotated screenshots AND written instructions/commands required



Install Nano Editor

Legible, annotated screenshots AND written instructions/commands required

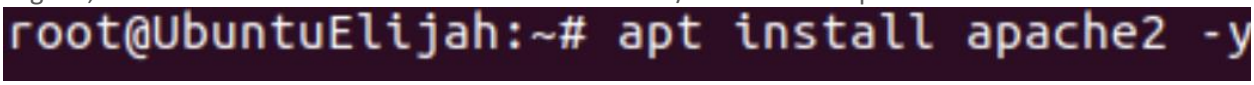


Install Git

[This step in the assignments may not be in order. Please reference all Week 2 assignmentrs for this step]
Legible, annotated screenshots AND written instructions/commands required

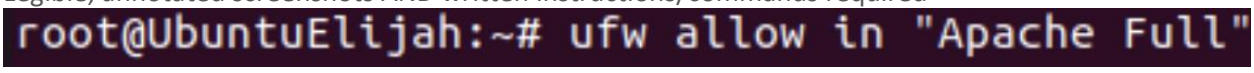
Install Apache2

Legible, annotated screenshots AND written instructions/commands required



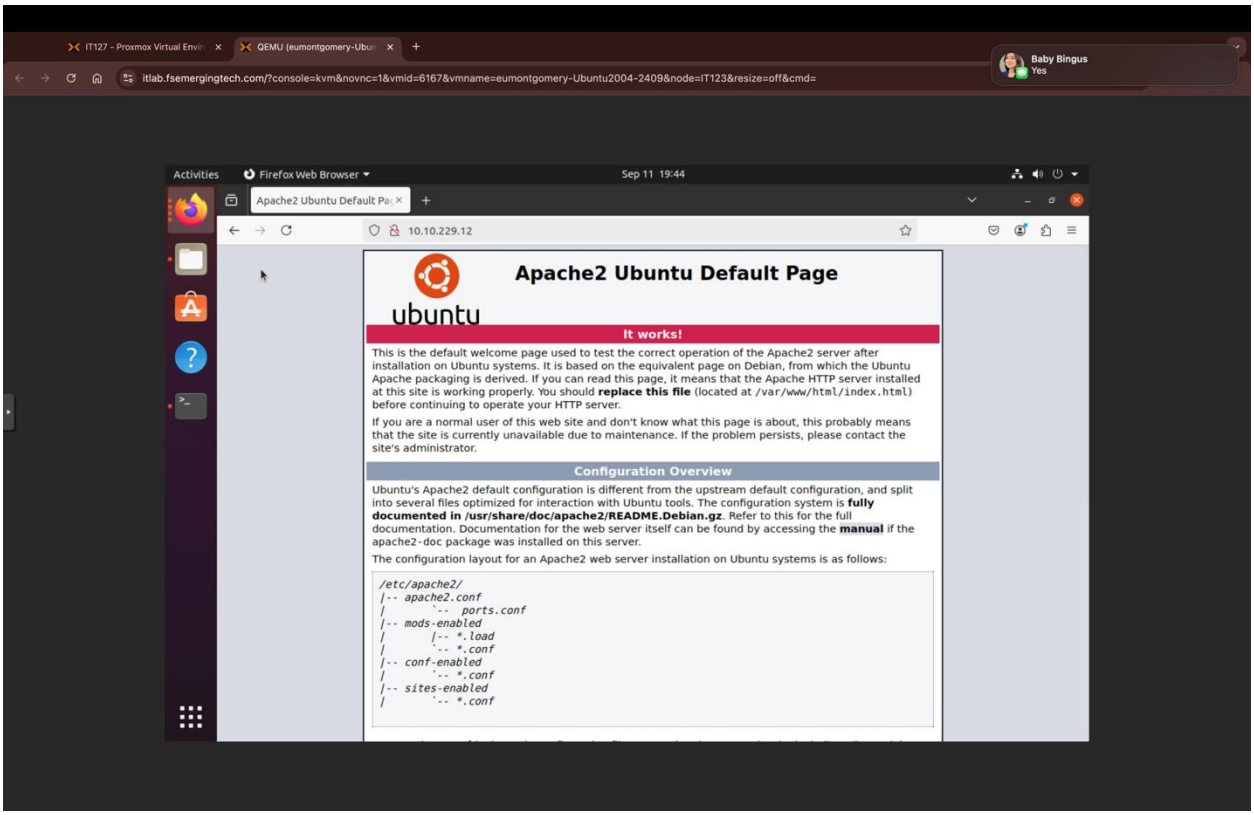
Open Firewall Ports 80 and 443

Legible, annotated screenshots AND written instructions/commands required



Browse to Apache2 Ubuntu Default Page

Legible, annotated screenshots AND written instructions/commands required



Install MySQL

Legible, annotated screenshots AND written instructions/commands required

```
root@UbuntuElijah:~# apt install mysql-server -y
```

Alter root user password (root@localhost)

Legible, annotated screenshots AND written instructions/commands required

NOTE: Document root user password in table at top of document.

```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'Fullsail1!';
```

Flush Privileges

Legible, annotated screenshots AND written instructions/commands required

```
mysql> FLUSH PRIVILEGES;
```

Exit MySQL

Legible, annotated screenshots AND written instructions/commands required

```
mysql> exit
```

Install PHP

Install Required PHP Libraries

Legible, annotated screenshots AND written instructions/commands required

```
root@UbuntuElijah:~# apt install php libapache2-mod-php php-mysql
```

Install Required MySQL Libraries

Legible, annotated screenshots AND written instructions/commands required

```
root@UbuntuElijah:~# apt install php-curl php-gd php-xml php-mbstring php-xmlrpc php-zip php-zip php-soap php-intl
```

Enable URL Rewrites (clean URLs)

Legible, annotated screenshots AND written instructions/commands required

```
root@UbuntuElijah:~# a2enmod rewrite
```

Restart Apache Service

Legible, annotated screenshots AND written instructions/commands required

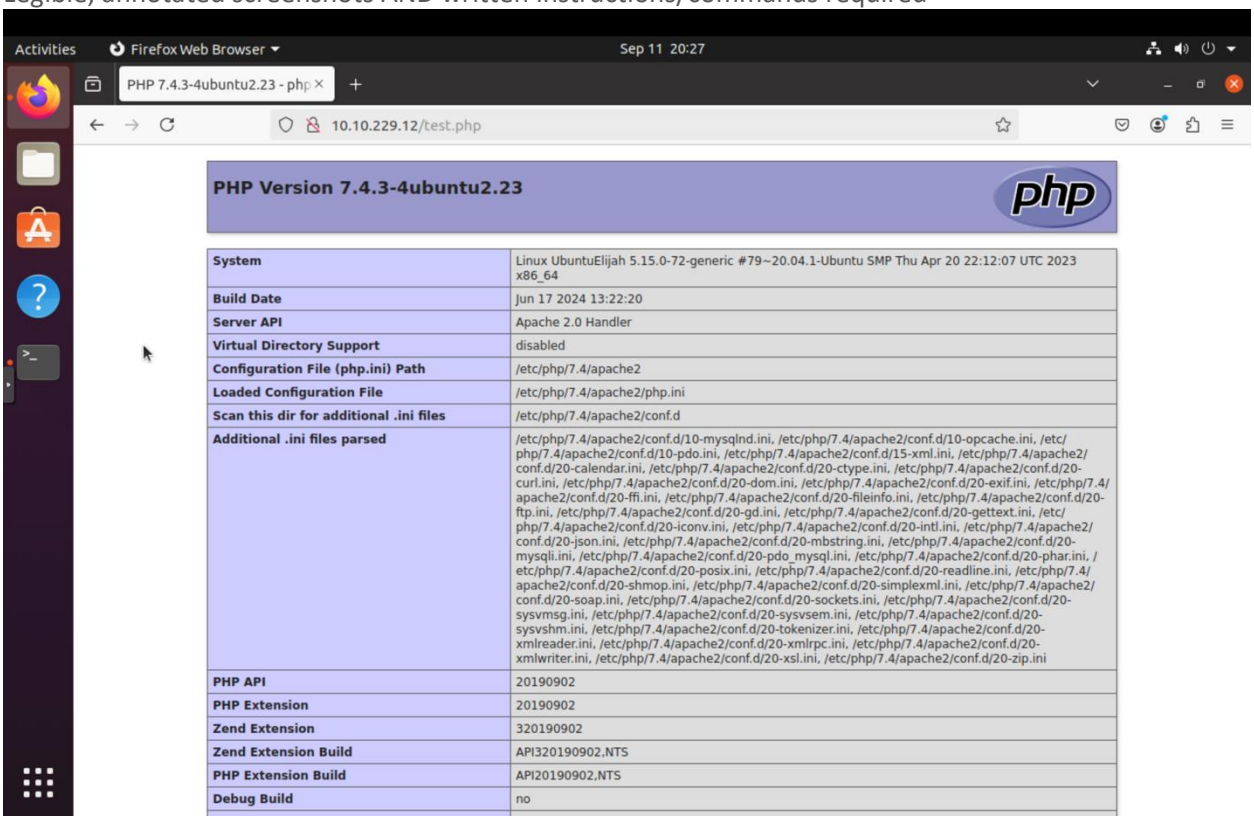
```
root@UbuntuElijah:~# systemctl restart apache2
```

Create a test.php Web Page

Legible, annotated screenshots AND written instructions/commands required

Test the test.php Web Page

Legible, annotated screenshots AND written instructions/commands required



Database Configuration in MySQL

Log into MySQL

Legible, annotated screenshots AND written instructions/commands required

```
root@UbuntuElijah:~# mysql -u root -p
```

Create MySQL WordPress Database (WordPressDB)

Legible, annotated screenshots AND written instructions/commands required

```
mysql> CREATE DATABASE WordpressDB DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci;
Query OK, 1 row affected (0.29 sec)
```

Create MySQL WordPress User (WordPressUser)

Legible, annotated screenshots AND written instructions/commands required

NOTE: Document WordPress User password in table at top of document.

```
mysql> CREATE USER 'WordPressUser'@'localhost' IDENTIFIED BY 'Fullsail1!';
```

Grant Privileges to the WordPress Database (WordPressDB) to WordPress User (WordPressUser)

Legible, annotated screenshots AND written instructions/commands required

```
mysql> GRANT ALL ON WordPressDB.* TO 'WordPressUser'@'localhost';
```

Flush Privileges

Legible, annotated screenshots AND written instructions/commands required

```
mysql> FLUSH PRIVILEGES;
```

Exit MySQL

Legible, annotated screenshots AND written instructions/commands required

```
mysql> exit;
```

Install WordPress

Grant Permission to /var/www/html/ Directory to WordPress User

Legible, annotated screenshots AND written instructions/commands required

```
root@UbuntuElijah:~# chown $USER:$USER /var/www/html/*
```

Delete Files from /var/www/html/ Directory

Legible, annotated screenshots AND written instructions/commands required

NOTE: Please remember that, when you delete the files from the /var/www/html/ directory, you will be deleting your test.php file from here.

```
root@UbuntuElijah:~# rm /var/www/html/*
```

Verify /var/www/html/ Directory is Empty

Legible, annotated screenshots AND written instructions/commands required

```
root@UbuntuElijah:/var/www/html# ls -la
```

Clone WordPress to /var/www/html/ Directory

Legible, annotated screenshots AND written instructions/commands required

```
root@UbuntuElijah:~# git clone https://github.com/WordPress/WordPress /var/www/html/
```

Verify /var/www/html/ Directory Contains WordPress Files

Legible, annotated screenshots AND written instructions/commands required

```
root@UbuntuElijah:~# ls -la /var/www/html
```

Verify Permissions on /var/www/html/ Directory

Legible, annotated screenshots AND written instructions/commands required

```
root@UbuntuElijah:~# ls -ls /var/www/html
```

Edit Ownership

Edit Ownership of and the contents of /var/www/html/ Directory

Legible, annotated screenshots AND written instructions/commands required

```
root@UbuntuElijah:~# chown -R www-data:www-data /var/www/html/*
```

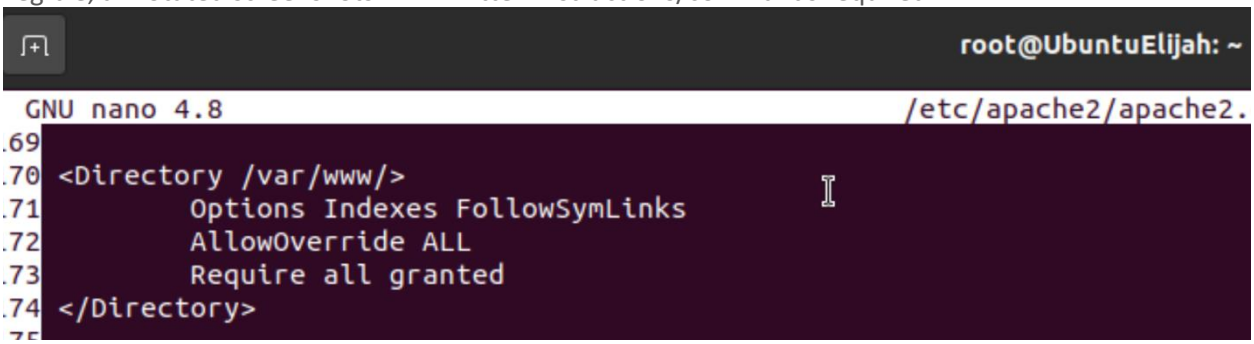
Edit the apache2.conf File

Legible, annotated screenshots AND written instructions/commands required

```
root@UbuntuElijah:~# nano /etc/apache2/apache2.conf --linenumbers
```

Override All Default Apache Directives

Legible, annotated screenshots AND written instructions/commands required



Create a .htaccess File in the /var/www/html/.git/ Directory

Legible, annotated screenshots AND written instructions/commands required

Restart the Apache Service

Legible, annotated screenshots AND written instructions/commands required

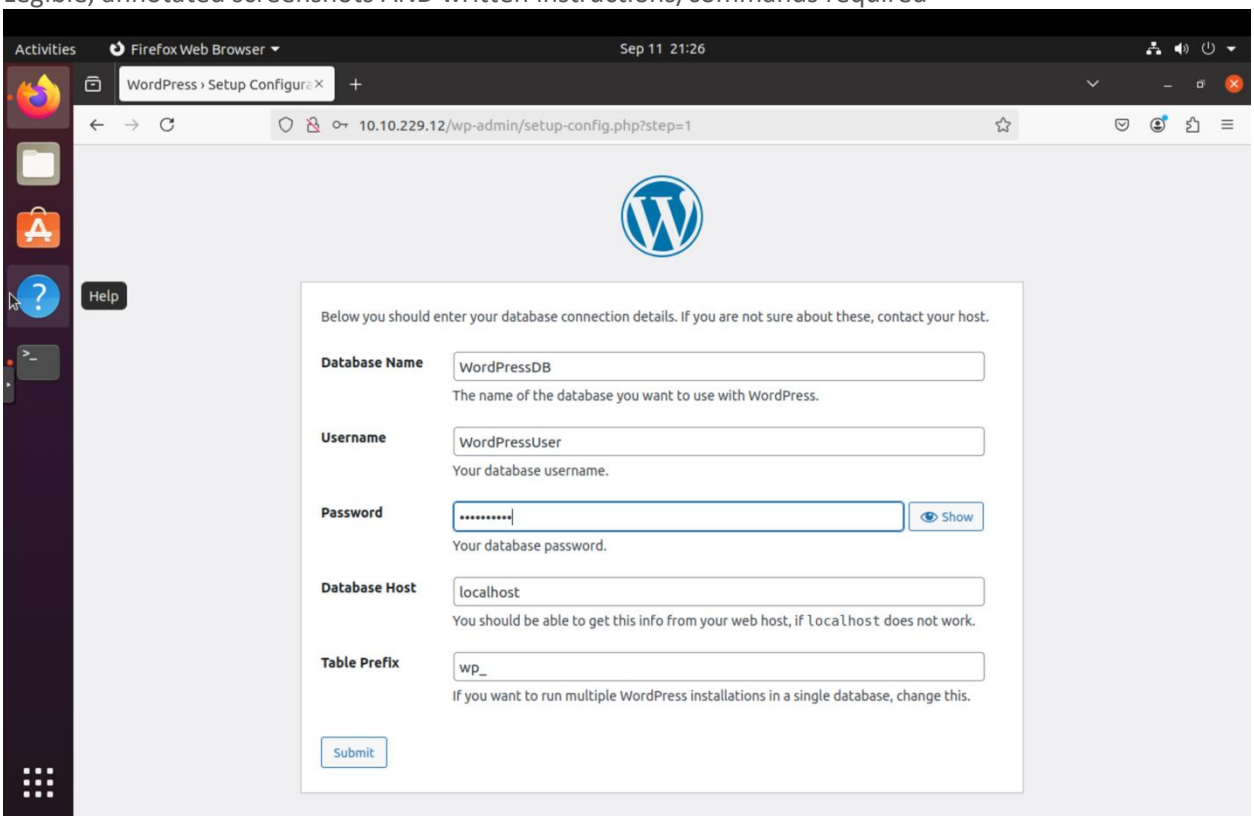
```
root@UbuntuElijah:~# systemctl restart apache2
```

WordPress Configuration

Configure WordPress

WordPress Configuration Selections

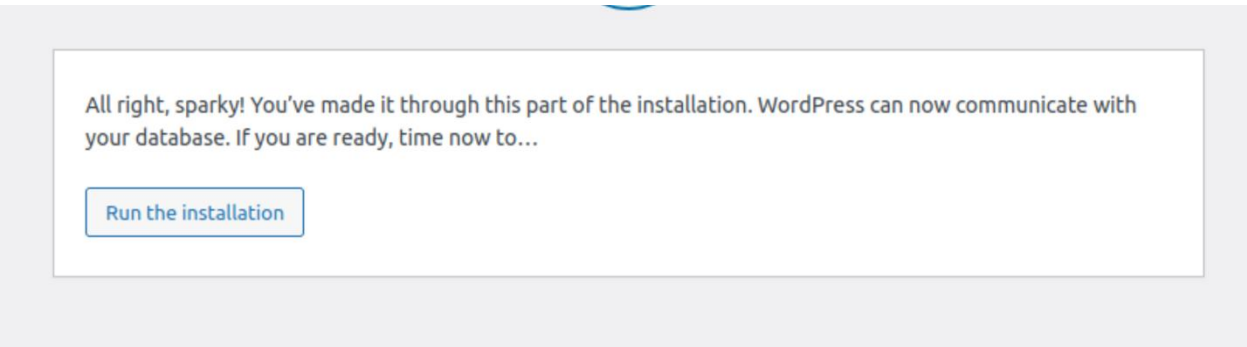
Legible, annotated screenshots AND written instructions/commands required



- **Database Name:** WordPressDB
- **Username:** WordPressUser
- **Password:** *this is your WordPressUser password*
- **Database Host:** localhost
- **Table Prefix:** wp_

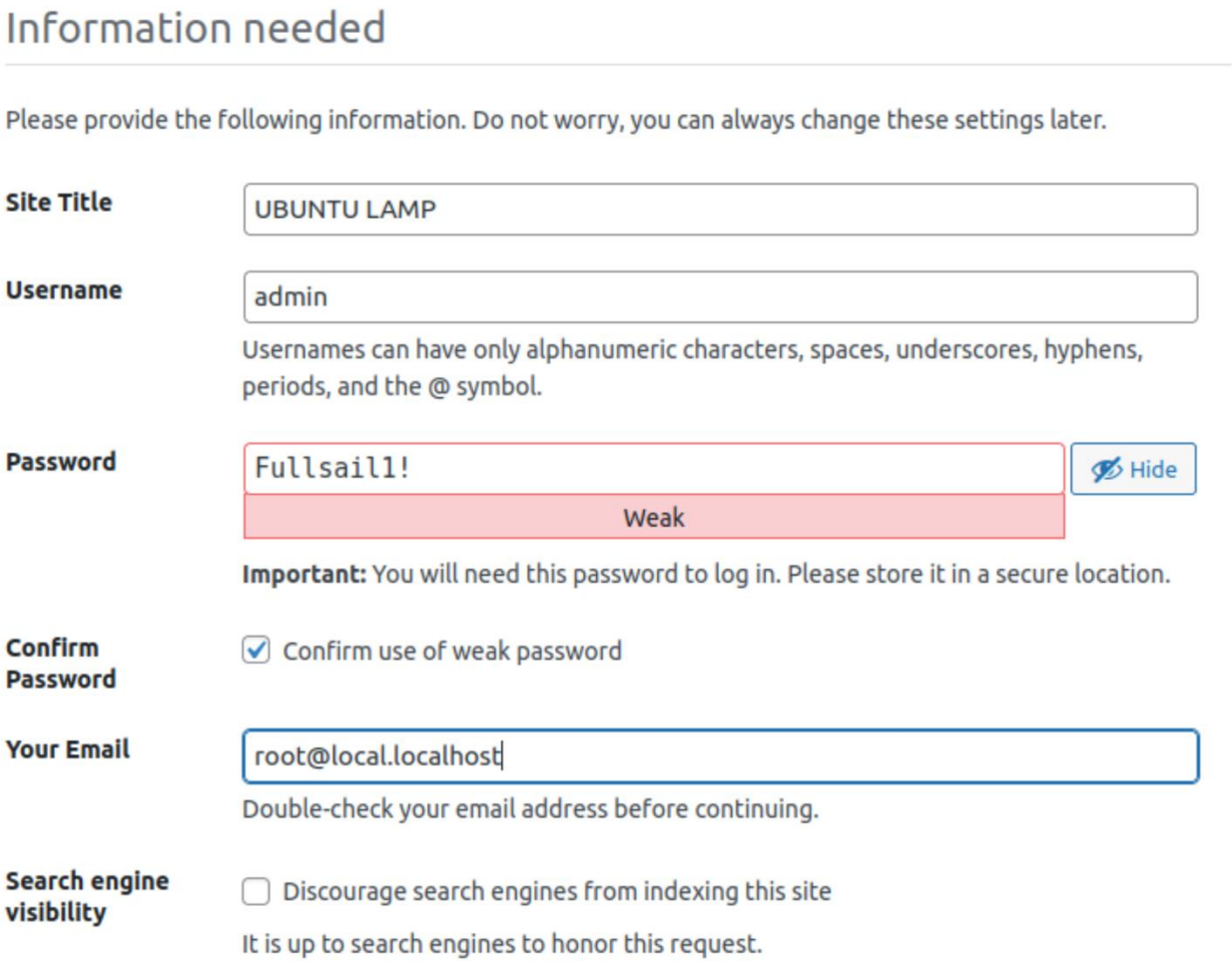
Run Installation

Legible, annotated screenshots AND written instructions/commands required



Create an Admin WordPress User

Legible, annotated screenshots AND written instructions/commands required
NOTE: Document WordPress admin user password in table at top of document.

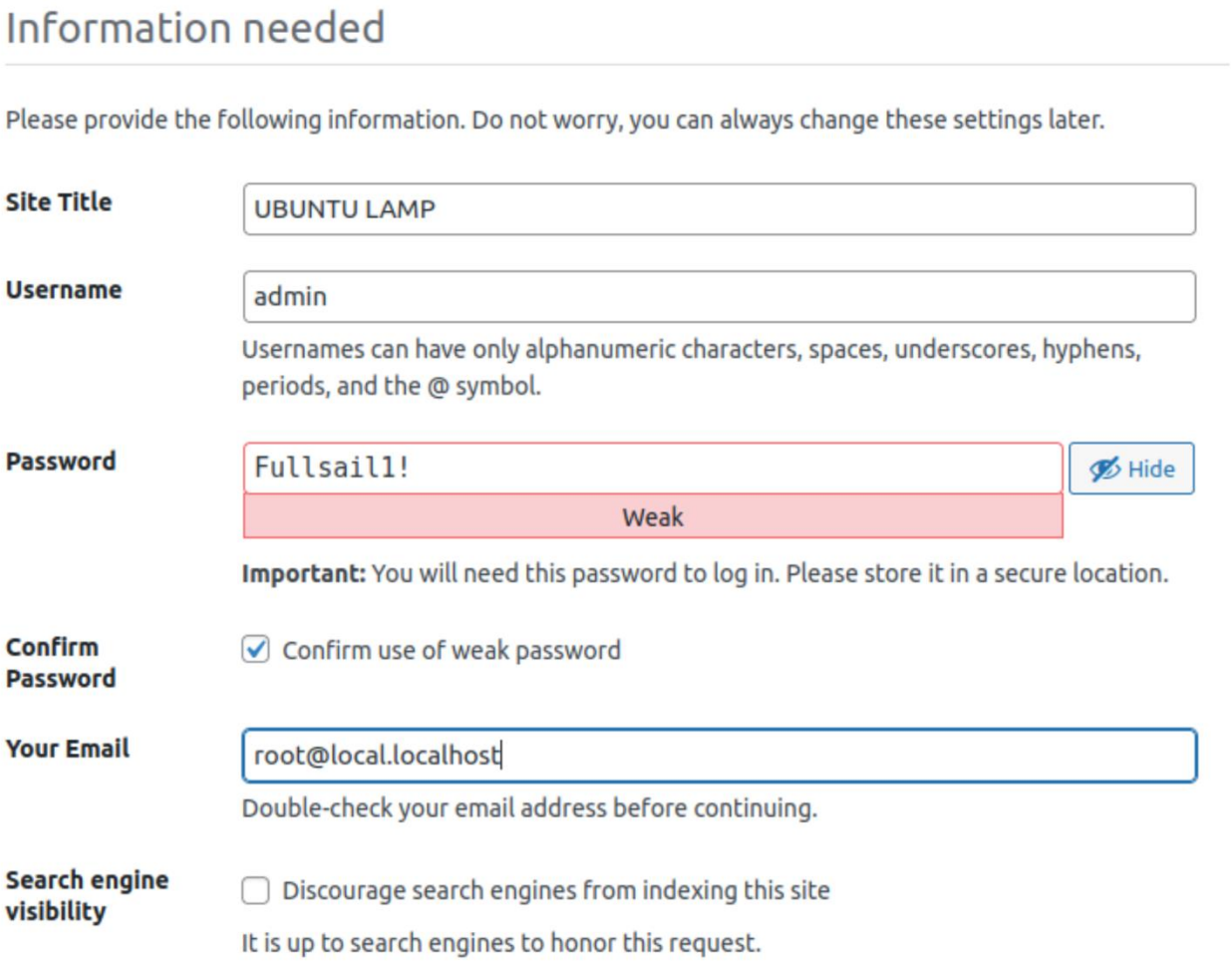


WordPress Site Selections

Legible, annotated screenshots AND written instructions/commands required

- **Site Title:** Ubuntu LAMP

- **Username**: admin
- **Password:** *You will create this*
- **Your email:** root@localhost.local
- **Search Engine Visibility:** leave unchecked



Test WordPress Website

Legible, annotated screenshots AND written instructions/commands required

This do be testin

Sep 12, 2024 — by admin in Uncategorized

Comments

One response to “This do be testin”

admin
September 12, 2024

He do be testin

END OF MILESTONE 2

WordPress Security Settings and Configurations

File Permissions

Before and After Legible, annotated screenshots AND written instructions/commands required

Vulnerability

```
root@UbuntuElijah:~# cd /var/www/html/ && ls -la
total 252
drwxr-xr-x  6 www-data www-data  4096 Sep 11 21:35 .
drwxr-xr-x  3 root     root      4096 Sep 11 19:31 ..
drwxr-xr-x  8 root     root      4096 Sep 11 21:20 .git
-rw-r--r--  1 www-data www-data   523 Sep 11 21:35 .htaccess
-rw-r--r--  1 www-data www-data   405 Sep 11 21:01 index.php
-rw-r--r--  1 www-data www-data 19915 Sep 11 21:01 license.txt
-rw-r--r--  1 www-data www-data  7409 Sep 11 21:01 readme.html
-rw-r--r--  1 www-data www-data  7387 Sep 11 21:01 wp-activate.php
drwxr-xr-x  9 www-data www-data  4096 Sep 11 21:01 wp-admin
-rw-r--r--  1 www-data www-data   351 Sep 11 21:01 wp-blog-header.php
-rw-r--r--  1 www-data www-data  2323 Sep 11 21:01 wp-comments-post.php
-rw-rw-rw-  1 www-data www-data  3322 Sep 11 21:31 wp-config.php
-rw-r--r--  1 www-data www-data  3033 Sep 11 21:01 wp-config-sample.php
drwxr-xr-x  5 www-data www-data  4096 Sep 11 21:35 wp-content
-rw-r--r--  1 www-data www-data  5617 Sep 11 21:01 wp-cron.php
drwxr-xr-x 30 www-data www-data 12288 Sep 11 21:01 wp-includes
-rw-r--r--  1 www-data www-data  2502 Sep 11 21:01 wp-links-opml.php
-rw-r--r--  1 www-data www-data  3937 Sep 11 21:01 wp-load.php
-rw-r--r--  1 www-data www-data 51198 Sep 11 21:01 wp-login.php
-rw-r--r--  1 www-data www-data  8525 Sep 11 21:01 wp-mail.php
-rw-r--r--  1 www-data www-data 28844 Sep 11 21:01 wp-settings.php
-rw-r--r--  1 www-data www-data 34385 Sep 11 21:01 wp-signup.php
-rw-r--r--  1 www-data www-data  4885 Sep 11 21:01 wp-trackback.php
-rw-r--r--  1 www-data www-data  3246 Sep 11 21:01 xmlrpc.php
```

Here, include a screenshot of the vulnerability you've identified before fixing the vulnerability. Include a brief description of what is occuring in the screenshot.

Configuration

In this step you will show the commands used to configure around or secure the screenshot. Include a brief description of what is occuring in the screenshot.

```
root@UbuntuElijah:/var/www/html# cd /var/www/html/ && ls -la
total 252
drwxr-xr-x  6 www-data www-data  4096 Sep 11 21:35 .
drwxr-xr-x  3 root     root      4096 Sep 11 19:31 ..
drwxr-xr-x  8 root     root      4096 Sep 11 21:20 .git
-rw-r--r--  1 www-data www-data   523 Sep 11 21:35 .htaccess
-rw-r-----  1 www-data www-data   405 Sep 11 21:01 index.php
-rw-r-----  1 www-data www-data 19915 Sep 11 21:01 license.txt
-rw-r-----  1 www-data www-data  7409 Sep 11 21:01 readme.html
-rw-r-----  1 www-data www-data  7387 Sep 11 21:01 wp-activate.php
drwxr-x---  9 www-data www-data  4096 Sep 11 21:01 wp-admin
-rw-r-----  1 www-data www-data   351 Sep 11 21:01 wp-blog-header.php
-rw-r-----  1 www-data www-data  2323 Sep 11 21:01 wp-comments-post.php
-rw-r-----  1 www-data www-data  3322 Sep 11 21:31 wp-config.php
-rw-r-----  1 www-data www-data  3033 Sep 11 21:01 wp-config-sample.php
drwxr-x---  5 www-data www-data  4096 Sep 11 21:35 wp-content
-rw-r-----  1 www-data www-data  5617 Sep 11 21:01 wp-cron.php
drwxr-x--- 30 www-data www-data 12288 Sep 11 21:01 wp-includes
-rw-r-----  1 www-data www-data  2502 Sep 11 21:01 wp-links-opml.php
-rw-r-----  1 www-data www-data  3937 Sep 11 21:01 wp-load.php
-rw-r-----  1 www-data www-data 51198 Sep 11 21:01 wp-login.php
-rw-r-----  1 www-data www-data  8525 Sep 11 21:01 wp-mail.php
-rw-r-----  1 www-data www-data 28844 Sep 11 21:01 wp-settings.php
-rw-r-----  1 www-data www-data 34385 Sep 11 21:01 wp-signup.php
-rw-r-----  1 www-data www-data  4885 Sep 11 21:01 wp-trackback.php
-rw-r-----  1 www-data www-data  3246 Sep 11 21:01 xmlrpc.php
root@UbuntuElijah:/var/www/html#
```

Validation

Finally, you will have a screenshot showing the vulnerability has been closed, or the steps taken to reduce the impact of said vulnerability. Include a brief description of what is occuring in the screenshot.

```
user@UbuntuElijah:~$ cd /var/www/html/wp-admin
bash: cd: /var/www/html/wp-admin: Permission denied
```

Securing wp-config.php

Before and After Legible, annotated screenshots AND written instructions/commands required

Vulnerability

Here, include a screenshot of the vulnerability you've identified before fixing the vulnerability. Include a brief description of what is occuring in the screenshot.

```
root@UbuntuElijah:~# cd /var/www/html/ && ls -la
total 252
drwxr-xr-x  6 www-data www-data  4096 Sep 11 21:35 .
drwxr-xr-x  3 root     root      4096 Sep 11 19:31 ..
drwxr-xr-x  8 root     root      4096 Sep 11 21:20 .git
-rw-r--r--  1 www-data www-data   523 Sep 11 21:35 .htaccess
-rw-r-----  1 www-data www-data   405 Sep 11 21:01 index.php
-rw-r-----  1 www-data www-data 19915 Sep 11 21:01 license.txt
-rw-r-----  1 www-data www-data  7409 Sep 11 21:01 readme.html
-rw-r-----  1 www-data www-data  7387 Sep 11 21:01 wp-activate.php
drwxr-x---  9 www-data www-data  4096 Sep 11 21:01 wp-admin
-rw-r-----  1 www-data www-data   351 Sep 11 21:01 wp-blog-header.php
-rw-r-----  1 www-data www-data  2323 Sep 11 21:01 wp-comments-post.php
-rw-r-----  1 www-data www-data  3322 Sep 11 21:31 wp-config.php
-rw-r-----  1 www-data www-data  3033 Sep 11 21:01 wp-config-sample.php
drwxr-x---  5 www-data www-data  4096 Sep 11 21:35 wp-content
-rw-r-----  1 www-data www-data  5617 Sep 11 21:01 wp-cron.php
drwxr-x--- 30 www-data www-data 12288 Sep 11 21:01 wp-includes
-rw-r-----  1 www-data www-data  2502 Sep 11 21:01 wp-links-opml.php
-rw-r-----  1 www-data www-data  3937 Sep 11 21:01 wp-load.php
-rw-r-----  1 www-data www-data 51198 Sep 11 21:01 wp-login.php
-rw-r-----  1 www-data www-data  8525 Sep 11 21:01 wp-mail.php
-rw-r-----  1 www-data www-data 28844 Sep 11 21:01 wp-settings.php
-rw-r-----  1 www-data www-data 34385 Sep 11 21:01 wp-signup.php
-rw-r-----  1 www-data www-data  4885 Sep 11 21:01 wp-trackback.php
-rw-r-----  1 www-data www-data  3246 Sep 11 21:01 xmlrpc.php
```

Configuration

In this step you will show the commands used to configure around or secure the screenshot. Include a brief description of what is occuring in the screenshot.

```
root@UbuntuElijah:/var/www/html# mv wp-config.php /var/www/
```

Validation

Finally, you will have a screenshot showing the vulnerability has been closed, or the steps taken to reduce the impact of said vulnerability. Include a brief description of what is occuring in the screenshot.

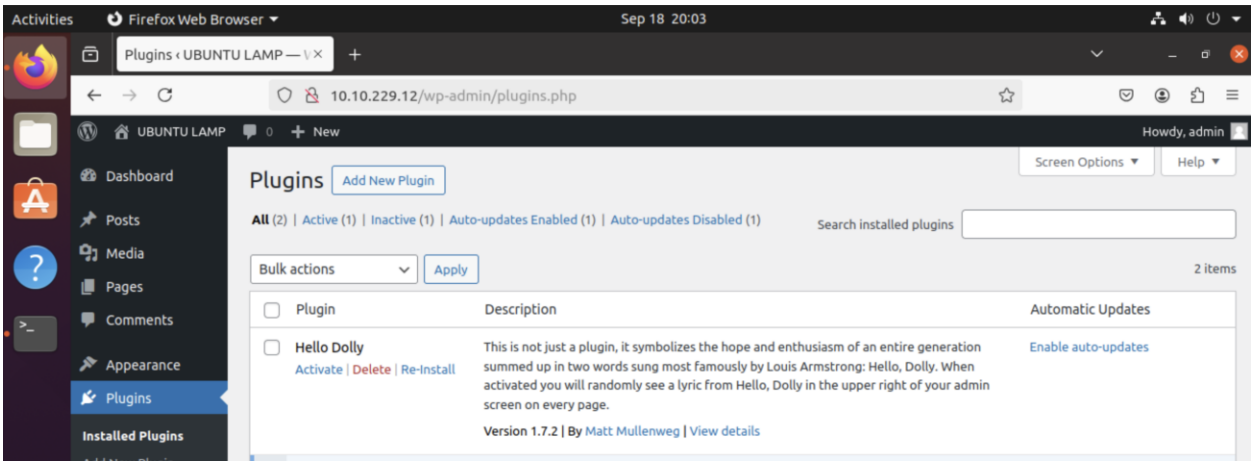
```
drwxr-xr-x  3 root     root     4096 Sep 18 19:39 .
drwxr-xr-x 15 root     root     4096 Sep 11 19:31 ..
drwxr-xr-x  6 www-data www-data 4096 Sep 18 19:39 html
-rw-r-----  1 www-data www-data 3322 Sep 11 21:31 wp-config.php
```

Firewall (Shield)

Before and After Legible, annotated screenshots AND written instructions/commands required

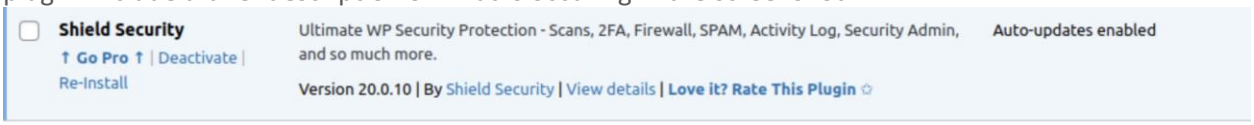
Vulnerability

Here, you will show a screenshot showing no Firewall Plugins installed at Layer 7 in the WordPress blog itself. Include a brief description of what is occuring in the screenshot.



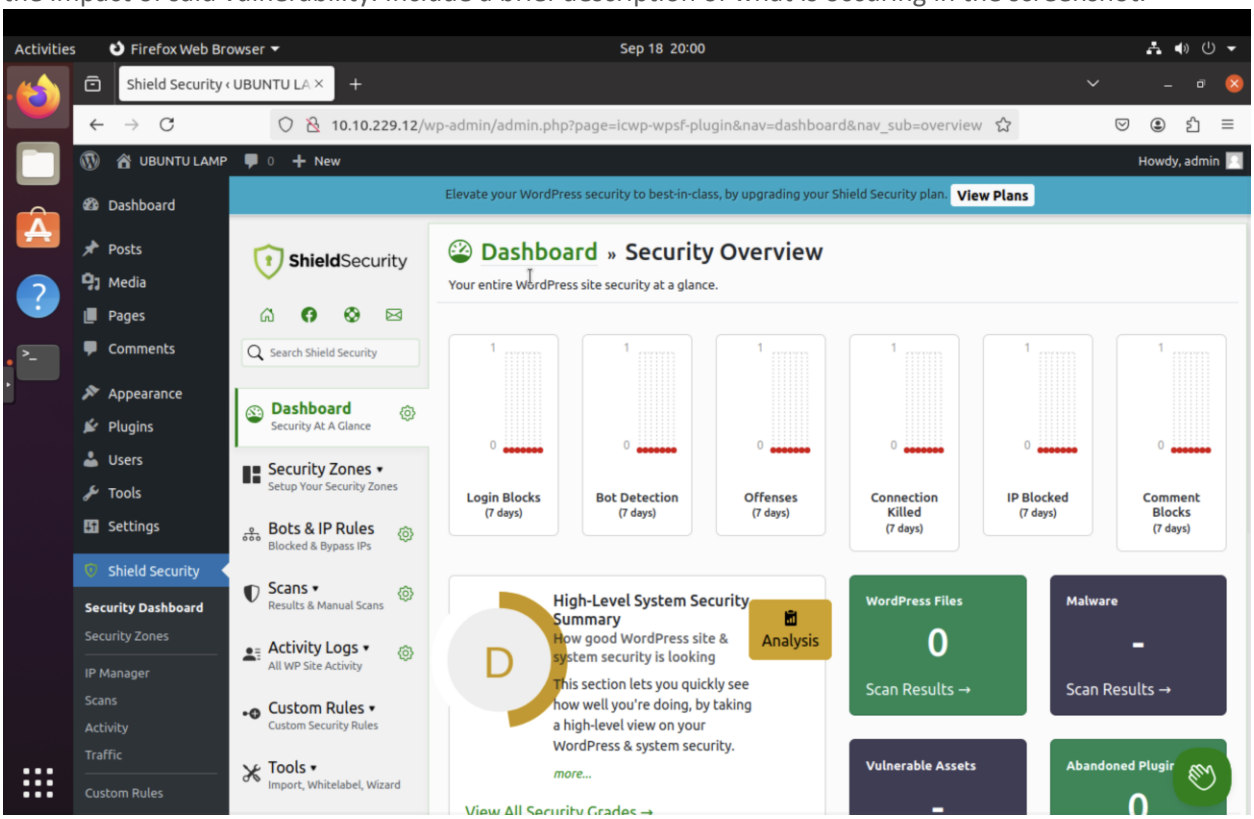
Configuration

In this step, include a screenshot of the installation, specifically the page where you may configure the plugin. Include a brief description of what is occuring in the screenshot.



Validation

Finally, you will have a screenshot showing the vulnerability has been closed, or the steps taken to reduce the impact of said vulnerability. Include a brief description of what is occuring in the screenshot.



Conclusion

Exemplary: HALF PAGE or 500-word conclusion (whichever comes first) fully summarizing your report. Your first paragraph should include a summary of what you secured on your WordPress blog site, and the steps taken to achieve this security. In your second paragraph, you will give a brief overview of defense-in-depth and cover how you applied defense-in-depth to the Milestone as a whole. Include how you can apply what you learned this month to installing, configuring, and securing other systems.

In this Proof of Concept for securing Mr. Marconi's WordPress site, I implemented multiple layers of protection to safeguard the system from potential vulnerabilities. The first step was to modify file permissions within critical WordPress directories: wp-admin, wp-content, and wp-includes. These directories house essential files and scripts that manage the site's functionality and appearance. By adjusting the permissions, I ensured that unauthorized users could not modify or delete important files, reducing the risk of accidental or malicious tampering. For instance, I set directory permissions to 755, allowing only the owner to make changes, while visitors and other users can only view the contents. Additionally, file permissions were restricted to 644, so only the owner can modify files, while others have read-only access.

The second step involved securing the WordPress configuration file, wp-config.php, which contains sensitive information such as database credentials and security keys. This file is highly valuable to attackers, so its default location within the web-accessible /var/www/html/ directory posed a significant risk. To mitigate this, I moved the file to a directory outside the publicly accessible web root, specifically to /var/www/. This step reduces the likelihood that an attacker could exploit a vulnerability in the WordPress site and gain access to the file. By relocating wp-config.php, I effectively protected the site's core configuration from exposure, even if other vulnerabilities are present.

The final security measure was the installation of a Layer 7 firewall plugin for WordPress. This firewall operates at the application layer, where most web-based attacks occur, including SQL injections, cross-site scripting (XSS), and brute force login attempts. By filtering and inspecting traffic at this layer, the firewall can block suspicious requests before they reach the WordPress backend, adding an essential layer of defense. I configured the firewall to automatically detect and block malicious IP addresses, limit login attempts, and prevent common attack patterns targeting WordPress sites. This configuration not only enhances the site's security but also reduces the risk of downtime due to hacking attempts or malicious traffic.

In applying the Defense-in-Depth strategy to this Proof of Concept, I took a layered approach to security, addressing multiple potential attack vectors. Defense-in-Depth is a cybersecurity principle that employs various security controls at different levels to create redundancy and ensure that if one defense mechanism fails, others are still in place to protect the system. In this case, file permissions prevent unauthorized access to sensitive files, the relocation of wp-config.php protects critical configuration data, and the Layer 7 firewall blocks malicious traffic at the application level. Each of these measures works in tandem to provide a robust defense against potential threats. Even if an attacker were able to bypass the firewall or exploit a vulnerability within WordPress, the other security layers would still prevent access to sensitive information or allow for detection before significant damage occurs.

The lessons learned in this Proof of Concept are applicable beyond the scope of this specific WordPress site. Understanding how to configure file permissions, secure sensitive configuration files, and implement firewall solutions is crucial for securing any web-based system or server. These skills can be applied to other content management systems (CMS), web applications, or networked environments where security is a top priority. Moving forward, I can use this knowledge to ensure that other systems I install or manage are secured with a layered approach, reducing the risk of exploitation and ensuring continued system integrity. The concept of Defense-in-Depth, in particular, will be a guiding principle as I apply these security practices to future projects.

Appendix A

NginX Config File

```
        location /blog {
          proxy_pass                          https://10.10.229.11:3001;
          proxy_set_header      Host          $http_host;
          proxy_set_header      X-Real-IP     $remote_addr;
          proxy_set_header      X-Forwarded-For $proxy_add_x_forwarded_for;
          proxy_read_timeout                  900;
        }
```

[Both Appendix A & B are required for Milestone 1. You may delete this after completing the Appendices.]

Appendix B

NginX Access Log File

```
[root@NginxMontgomery ~]# tail /var/log/nginx/access.log
10.10.229.12 - - [10/Sep/2024:14:17:28 -0400] "GET /blog HTTP/1.1" 404 3332 "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0) G
ecko/20100101 Firefox/113.0" "-"
10.10.229.12 - - [10/Sep/2024:14:17:29 -0400] "GET /nginx-logo.png HTTP/1.1" 200 368 "http://10.10.229.10/blog" "Mozilla/5.0 (X11; U
buntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/113.0" "-"
10.10.229.12 - - [10/Sep/2024:14:17:29 -0400] "GET /poweredby.png HTTP/1.1" 200 1800 "http://10.10.229.10/blog" "Mozilla/5.0 (X11; U
buntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/113.0" "-"
10.10.229.12 - - [10/Sep/2024:14:17:29 -0400] "GET /favicon.ico HTTP/1.1" 404 3332 "http://10.10.229.10/blog" "Mozilla/5.0 (X11; Ubu
ntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/113.0" "-"
```

top

NginX Error Log File