|  |  |
| --- | --- |
| Name: | Muskan Shukla |
| Lab User ID: | 23SEK3324\_U14 |
| Date: | 10-01-2024 |
| Application Name: | dvwps: Damn Vulnerable WordPress Site |

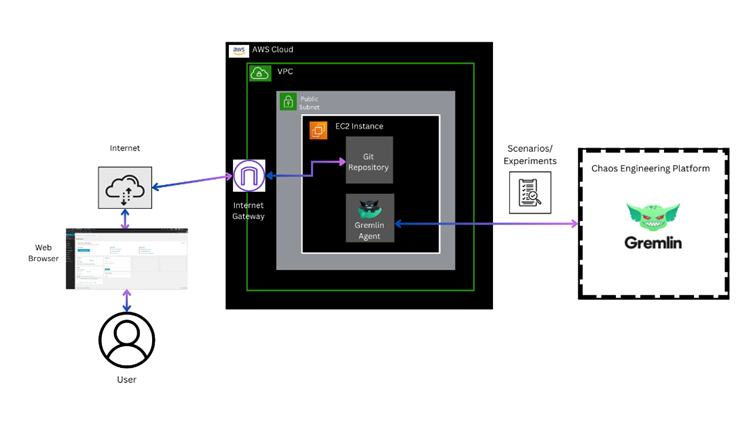
**Follow the below guidelines:**





System Architecture:

(Understand the system and document the physical and logical architecture of the system, use the shapes and icons to capture the system architecture)



Define system’s normal behavior:

(Define the steady state of the system is defined, thereby defining some measurable outputs which can indicate the system’s normal behavior)

The Damn Vulnerable WordPress Site (DVWS) is a WordPress website that is intentionally developed with various vulnerabilities. It's designed to be a fun and useful tool for WordPress developers, security researchers, and enthusiasts to learn, test, and experiment with WordPress security.

As such, its normal behavior involves being susceptible to different types of attacks and exploits that a secure WordPress website would not be. These may include SQL injection, Cross-Site-Scripting (XSS), and Cross-Site Request Forgery (CSRF), among others.

It may allow users to easily gain administrative access, alter site content, or release sensitive data. It's used specifically in a controlled environment to learn how to reinforce the security of a WordPress site when building one or to practice skills of ethical hacking.

Hypothesis:

(During an experiment, we need a hypothesis for comparing to a stable control group, and the same applies here too. If there is a reasonable expectation for a particular action according to which we will change the steady state of a system, then the first thing to do is to fix the system so that we accommodate for the action that will potentially have that effect on the system. For eg: "If one of our database servers fails, our service will automatically switch to a backup server, and users will not experience any downtime or data loss.")



**Known**

If a new plugin is added to the WordPress setup, the system will continue to function without critical vulnerabilities.Regular security scans will detect and flag any potential vulnerabilities introduced by the new plugin.

If there is a sudden increase in web traffic, the system will handle the load efficiently, and users will experience minimal latency

**Unknown**

**Unknown**

**Known**

If an unexpected system update is applied, the system will recover gracefully, and users won't experience service disruptions.System logs will show successful update installations without error spikes in user access or application errors.

If the internet gateway configuration is modified, the web application will remain accessible to users without interruptions. Network monitoring tools will show stable connectivity between the EC2 instance and the internet.

Experiment:

(Document your Preparation, Implementation, Observation and Analysis )

**Preparation :-**

Establish an AWS account if you don’t already have one. Log in to the AWS Management Console and access the EC2 Dashboard.

Create an AWS t2.medium instance. Choose your desired OS (e.g. Ubuntu Server 20.04 LTS).

After the instance is successfully launched, connect to your instance via SSH through the command line or use an SSH client.

**Repository :- https://github.com/vianasw/dvwps**

**Implementation :-**

Install Docker: Run the installation commands for Docker as ‘sudo apt install docker.io -y’

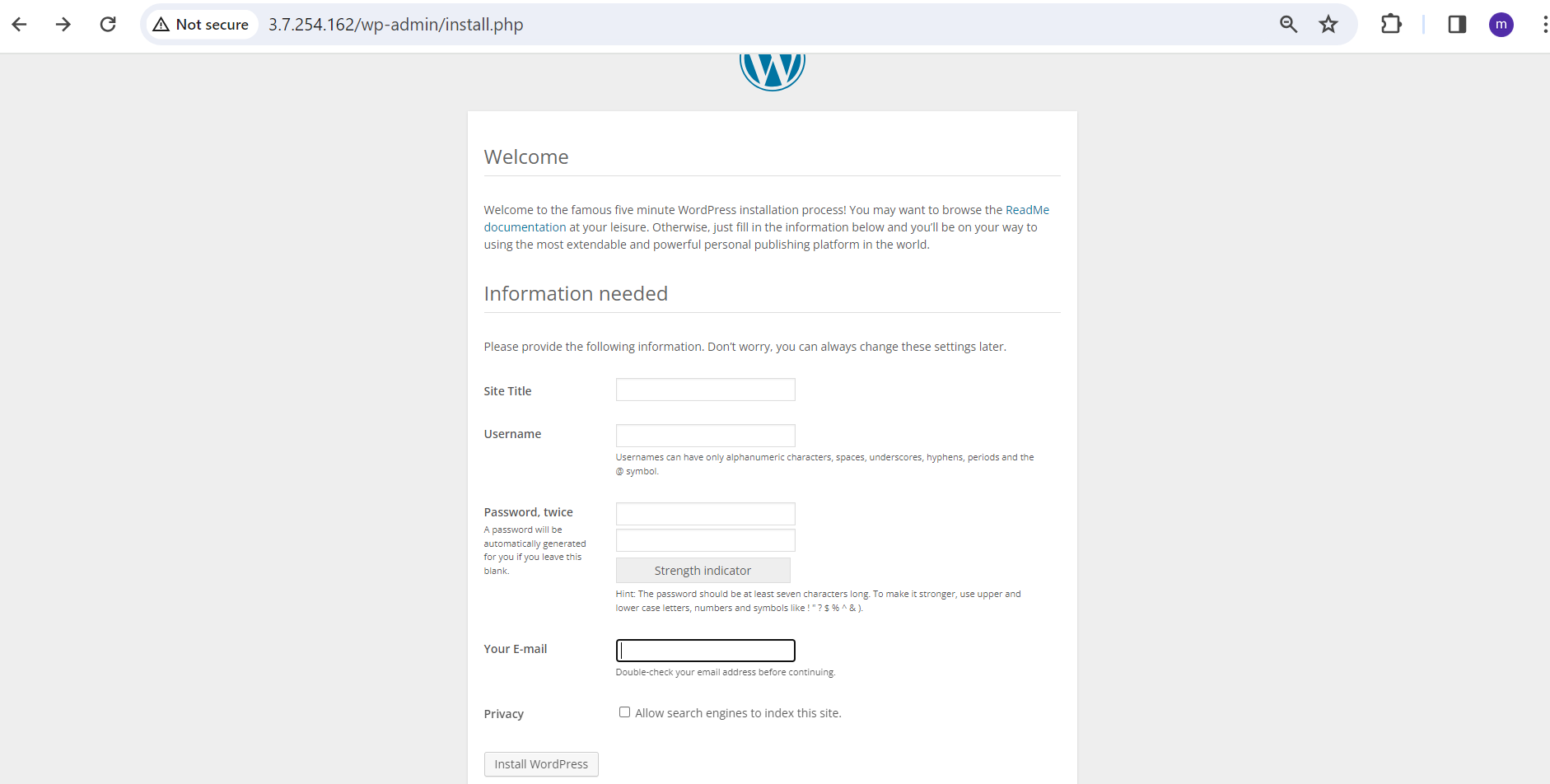
Docker's successful installation was confirmed by running the "docker --version" command

***Command –***

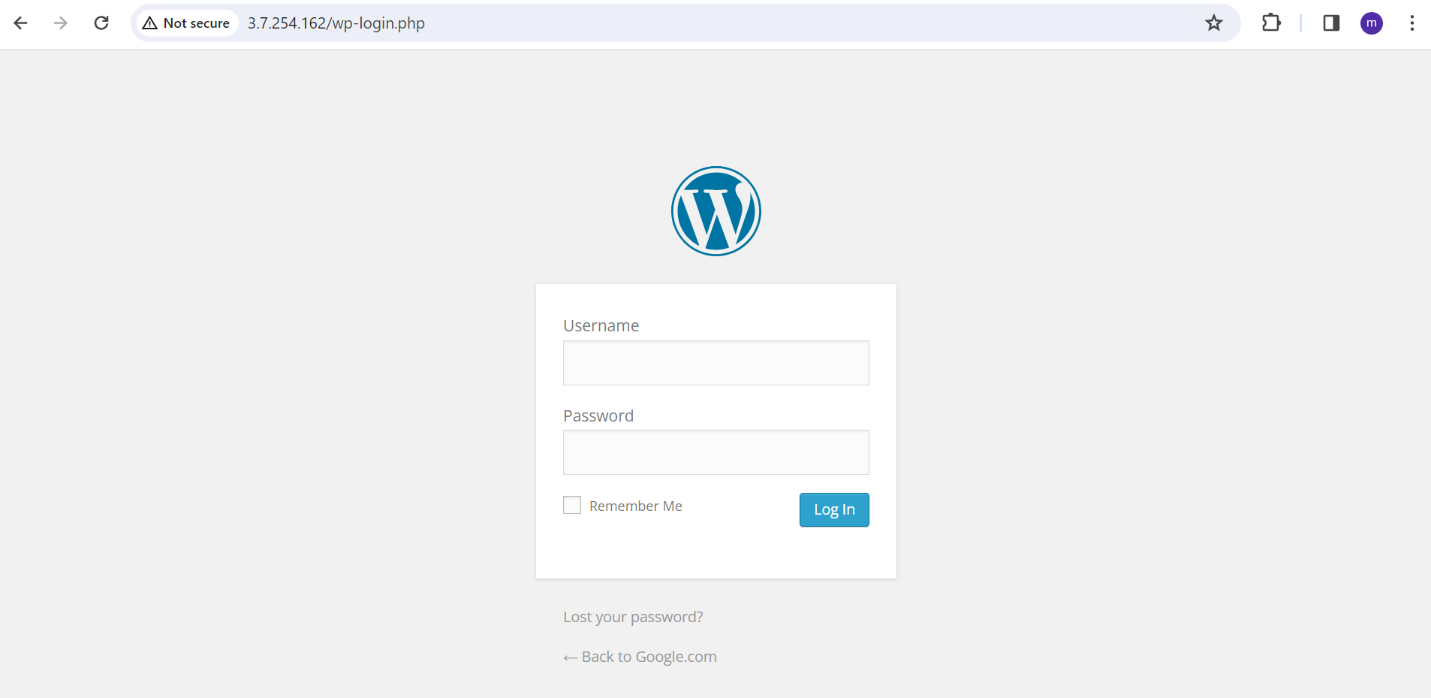
docker build -t username/mysql\_datastore .

docker run --name wp\_data username/mysql\_datastore

Live Webserver on browser

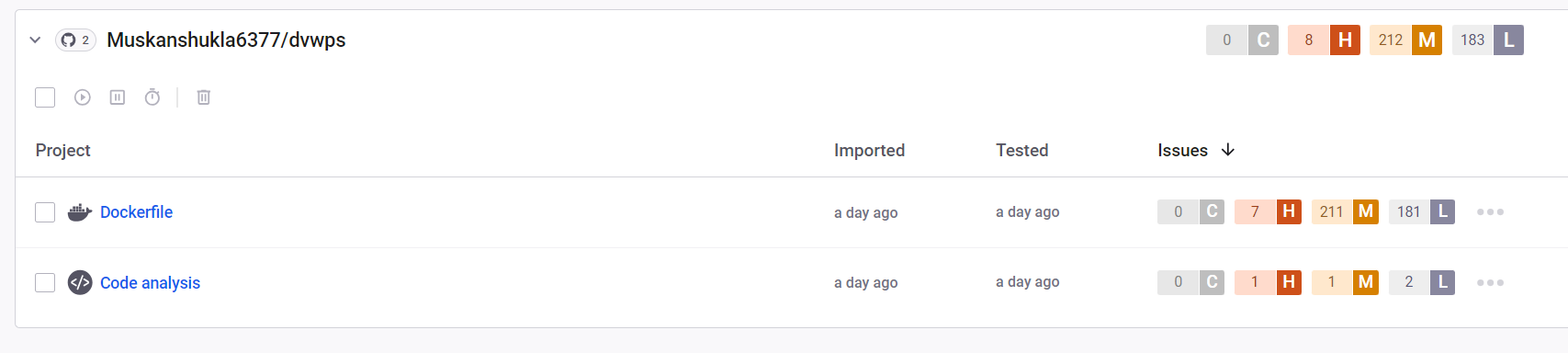


After filling all the details, login page is open where we have to write username and password



**Observation And Analysis :-**

Using SNYK, We have several vulnerability like 212 medium, 8 high and 183 low vulnerability



--> ***CVE-2015-3183 -*** The product receives input or data, but it does not validate or incorrectly validates that the input has the properties that are required to process the data safely and correctly.

-->***CVE-2021-3156*** – It is a heap-based buffer overflow vulnerability in Sudo, which is used in almost every UNIX or Linux based system. The security bug can allow an unprivileged user to gain root privileges in a vulnerable host.

-->[***CVE-2018-2767***](https://nvd.nist.gov/vuln/detail/CVE-2018-2767) ***-*** Difficult to exploit vulnerability allows low privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized read access to a subset of MySQL.