|  |  |
| --- | --- |
| Name: | Mazhar Hussain C K |
| Lab User ID: | 23SEK3324\_U06 |
| Date: | 10-01-2024 |
| Application Name: | 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)

WordPress

Container

MySQL

Container

Docker Engine

EC2 Instance

Two container are created in the docker one for MySQL as database for the wordpress site,and one for the wordpress site itself,with the linked database

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)

In the context of dvwps also called Damn Vulnerable Wordpress Site deployed on docker in ec2 instances works well

In normal conditions,the website is able to access through any ip,without giving having large latency.it is working

Under normal conditions in which large amount of data does not need to be transferred

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**

It contains intentional vulnerabilities.It is capable of handling low latency tasks

Things we are aware of but don’t understand.

942236

**Unknown**

**Unknown**

**Known**

Things we are neither aware of nor understand.

Things we understand but are not aware of.

We are going to conduct chaos experiment on this and we hypothesis the following outcomes

Latency attack:if we were to add latency to the website through chaos engineering we would still be able to access

It without any problem or denial of service

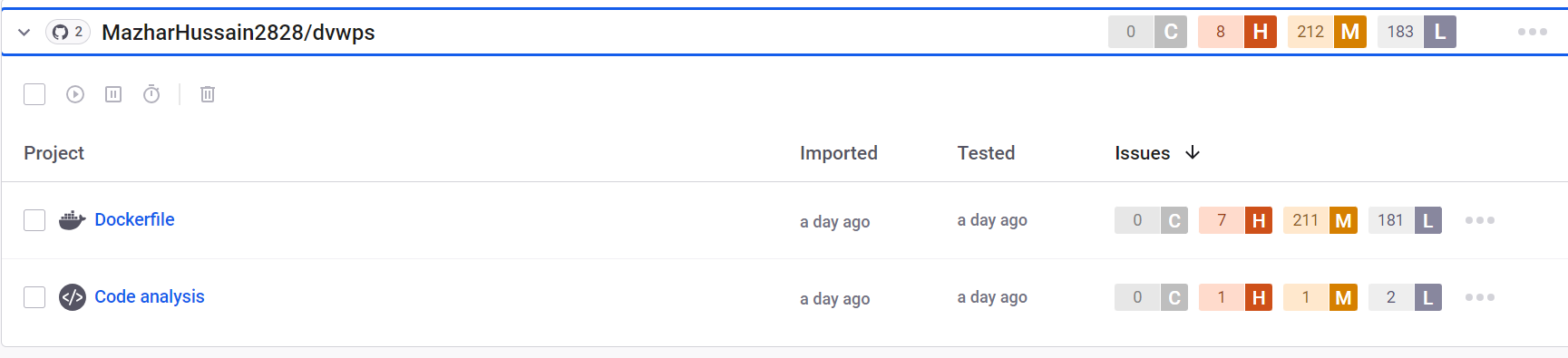
Also we intent to find vulnerabilities as this is an intentionaly vulnerable app

Experiment:

(Document your Preparation, Implementation, Observation and Analysis )

This project is done in aws ec2 instance with ubuntu image.Before hosting the project we used snyk for scanning the

Repository.the output for it is given below



As you can see most the errors arise from the docker file that has been used .Some of the main vulnerabilities are

Exposure of the resurce to wrong sphere which leads to another vulnerability leading to Improper Encoding or

Escaping of Output

Setup

First we cloned the repository to our ubuntu machine and installed docker in the machine

After that created a data only container Dockerfile for the mySQL that is going to be created

FROM ubuntu:14.04

MAINTAINER mazhar

VOLUME /var/lib/mysql

CMD ["true"]

Then build the image and run the container using

$ docker build -t username/mysql\_datastore .

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

After that we run the MySQL Docker

$ docker run --name wp\_db -p 3306:3306 --volumes-from wp\_data -v /path/to/dvwps/configs/:/etc/mysql/conf.d -e MYSQL\_ROOT\_PASSWORD=password -e MYSQL\_USER=wordpressuser -e MYSQL\_PASSWORD=password -e MYSQL\_DATABASE=wordpress -d mysql:5.6

This Command takes the config file for mySQL from the cloned repository and a mySQL container gets created

After that we specify the command for creating Wordpress Website

$ docker run -d --link wp\_db:mysql -p 80:80 -e DB\_NAME='wordpress' -e DB\_USER='wordpressuser' -e DB\_HOST='db\_host\_ip\_address' -e DB\_PASSWORD='password' vianasw/dvwps

The Website is now accessible via browser

A screenshot of a computer

Description automatically generated

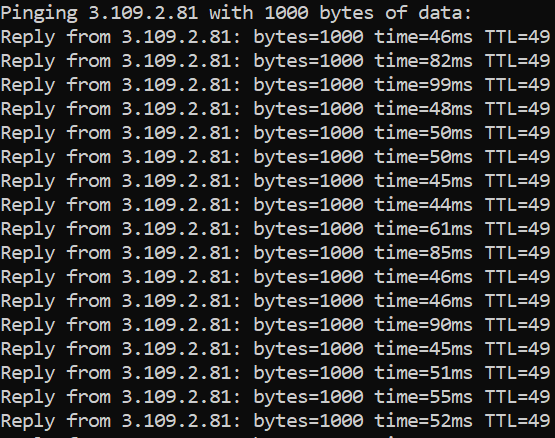
after configuration of username and password we prepare it for chaos xxperiment.for the chaos experiment we are

going to do latency attack on the wordpress website.upon successful installation of gremlin agent we ready it for

the attack,for monitoring purpose we used ping command to check the latency of the website during attack

we did a 100% latency attack on the wordpress website resulting in high latency.

Before Attack



Before the attack the latency is around 40-70 range and website was easily accessible through the web browser

During Attack

A screen shot of a computer screen

Description automatically generated

As you can see from this the latency of it increased +100 for the website.even with this much latency we were still

Able to access our website without having any issue main reason for this is the website being a lite one without a

Large number of media things which may increase the loading time of website.