Capture The Flag (CTF) Challenge using Docker - Web Application Vulnerability Lab

Creator: Dharanidharan V

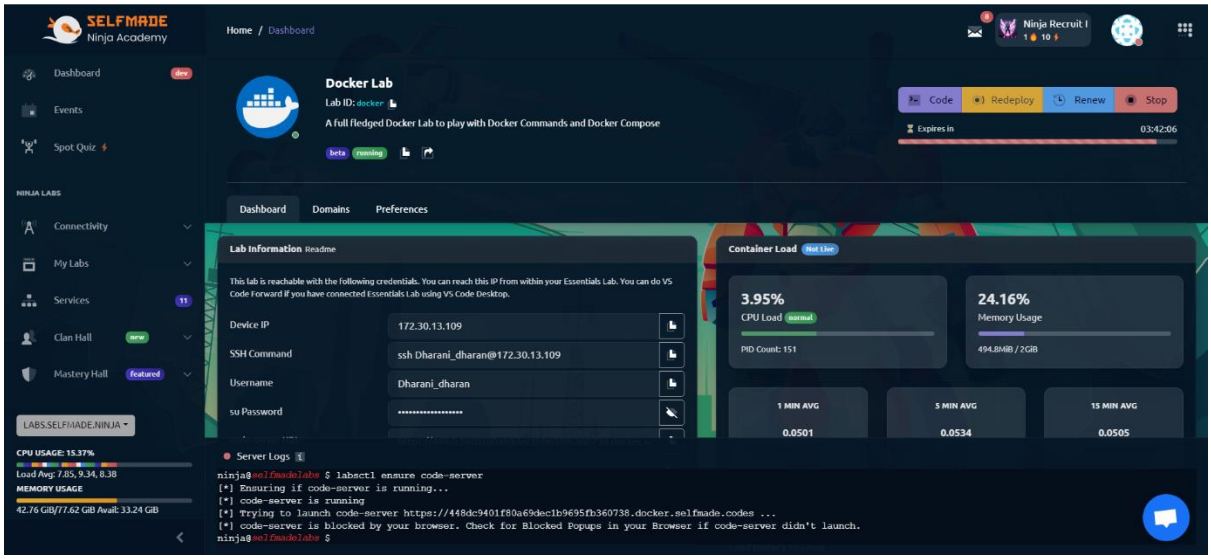
Overview

This document provides clear setup and run instructions for the Dockerized Web Application Vulnerability Lab. The lab contains four different challenges designed to test various web application vulnerabilities: SQL Injection, Cross-Site Scripting (XSS), Insecure File Upload, and Command Injection. Each challenge includes a description and a hint to help participants discover and exploit the vulnerabilities.

Setup Instructions

1. Environment:

Selfmade ninja's Docker lab is used in my assignment to build the CTF-Challenge Website



2. **Build the Docker Image:**

bash

docker build -t ctf-image .

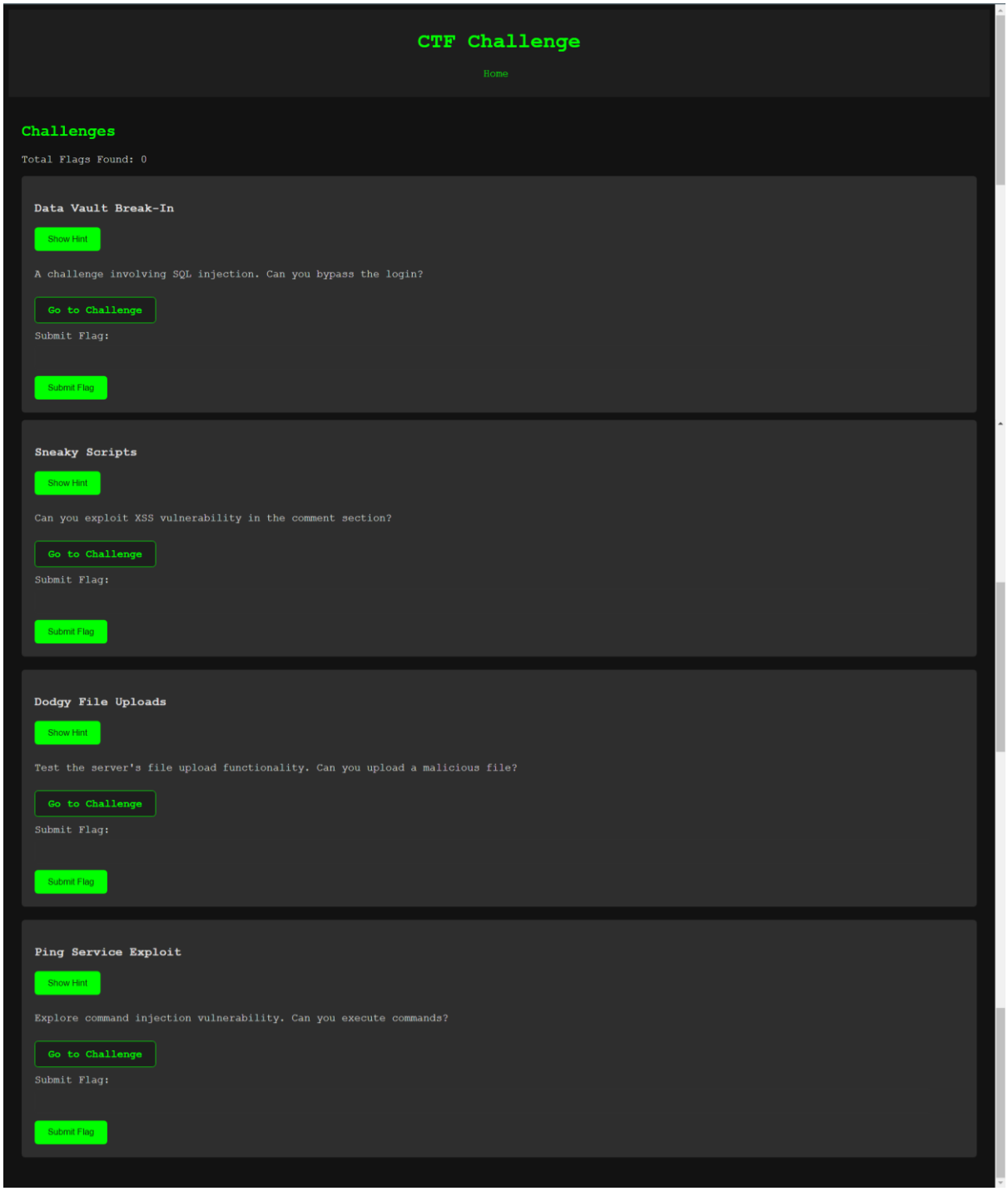
3. **Run the Docker Container:**

bash

docker run -d –name ctf-container -p 5000:5000 ctf-image

Once the Docker container is running, you can access the application through your web browser. The home page will provide links to each of the four challenges.

Challenges Home Page:



Challenges Description and Hints

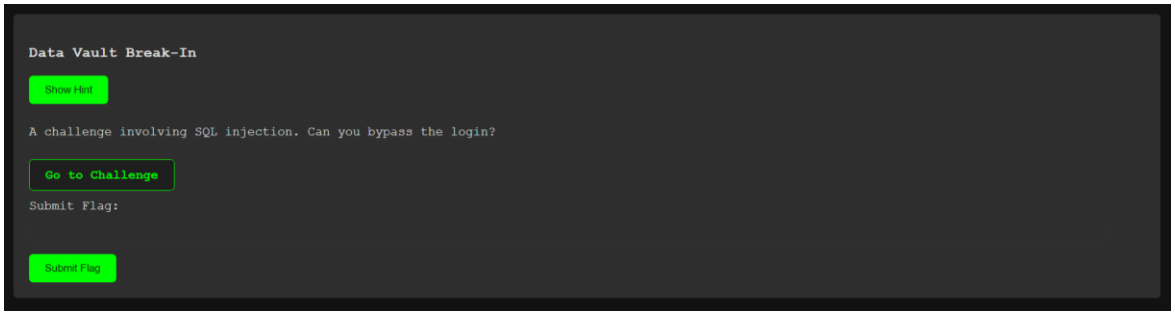
1. Data Vault Break-In (SQL Injection)

Description:

This challenge involves exploiting an SQL injection vulnerability to bypass the login page. Participants need to craft an SQL query to log in as an administrator.

Hint:

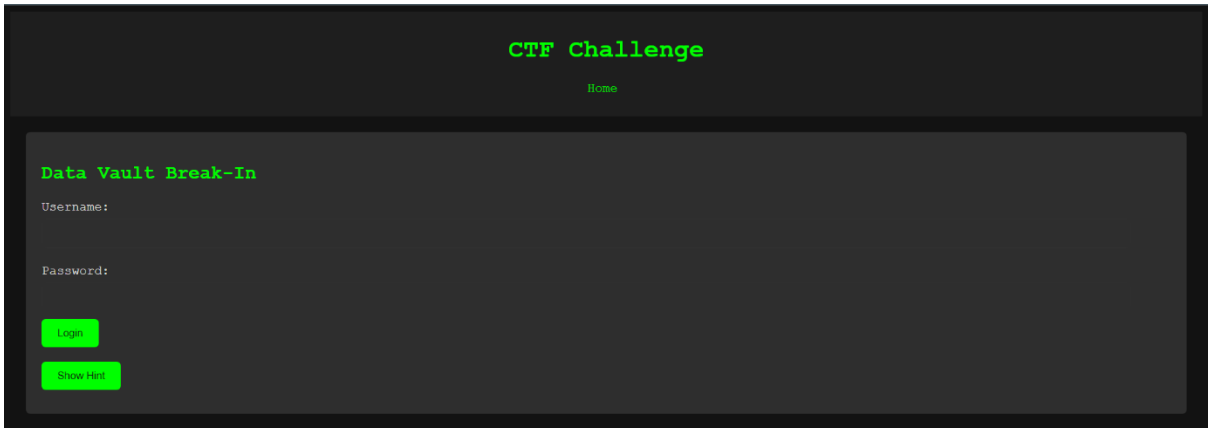
"Sometimes databases trust user input too much..."



Steps to Complete:

1. Go to the "Data Vault Break-In" challenge page.

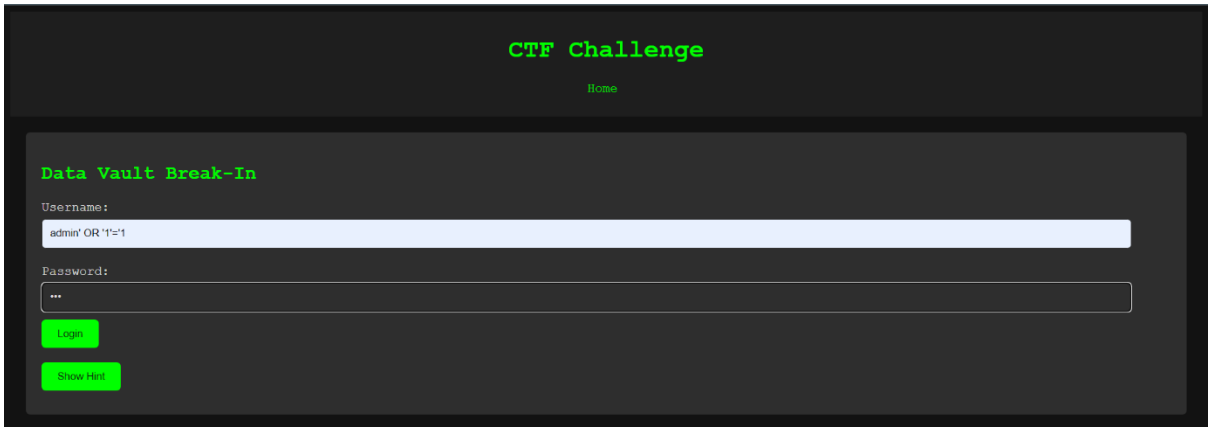
2. Attempt to log in using SQL injection techniques.



Example:

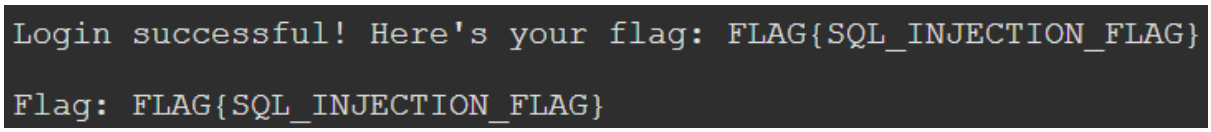
username: admin ' OR '1'='1

password: anything



Flag:

FLAG{SQL_INJECTION_FLAG}



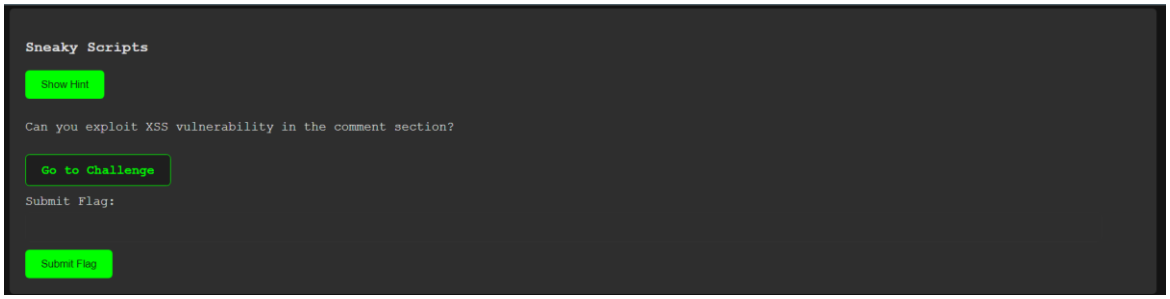
2**. Sneaky Scripts (Cross-Site Scripting - XSS)**

Description:

This challenge involves finding an XSS vulnerability in the comment section of a page. Participants need to insert a script into the comment box to execute a script.

Hint:

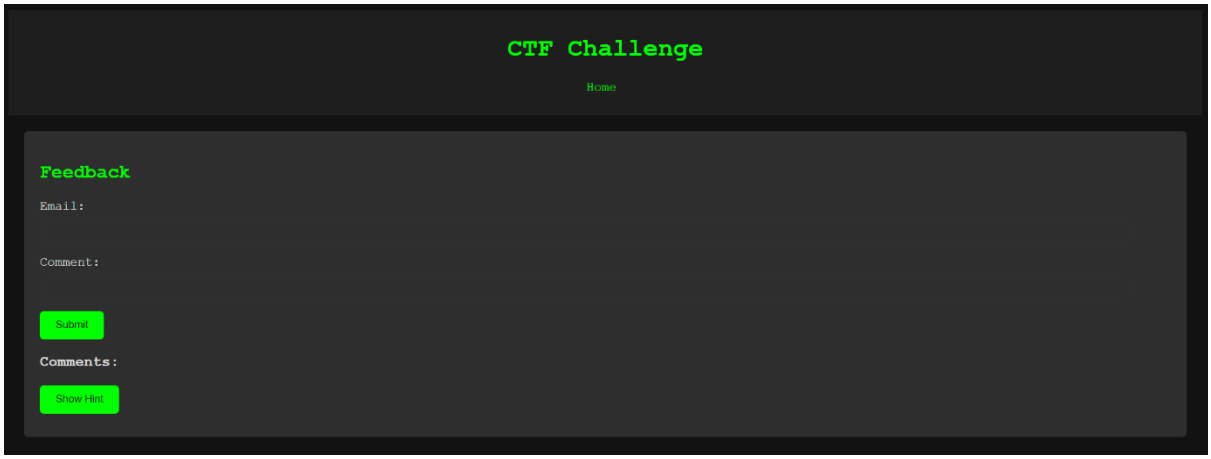
"Be careful what you let others post on your site..."



Steps to Complete:

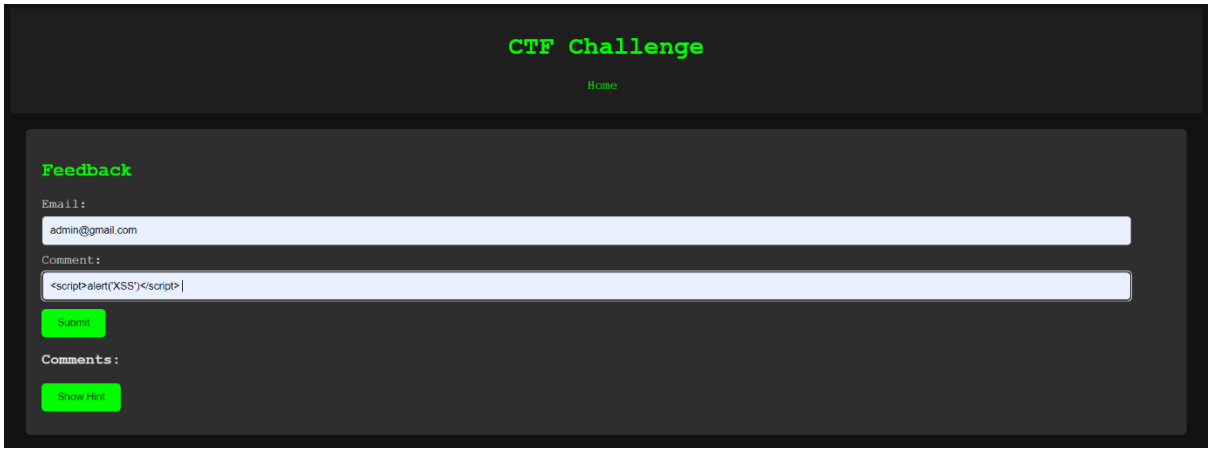
1. Go to the "Sneaky Scripts" challenge page.

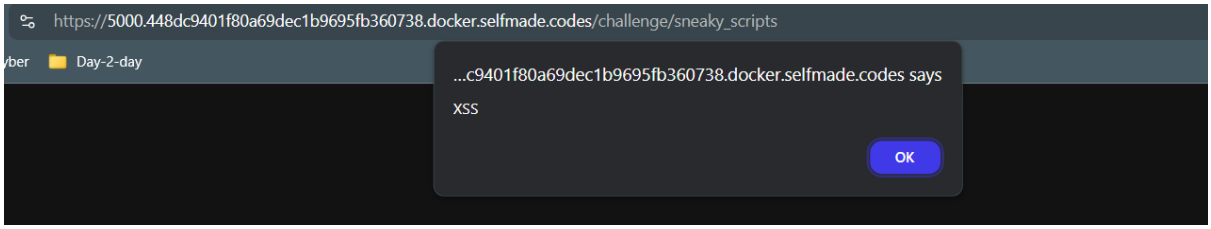
2. Post a comment containing a script tag.



Example:

<script>alert('XSS')</script>





Flag:

FLAG{XSS_FLAG}

```
Comment posted and XSS detected! Here's your flag: FLAG{XSS_FLAG}

Flag: FLAG{XSS_FLAG}
```

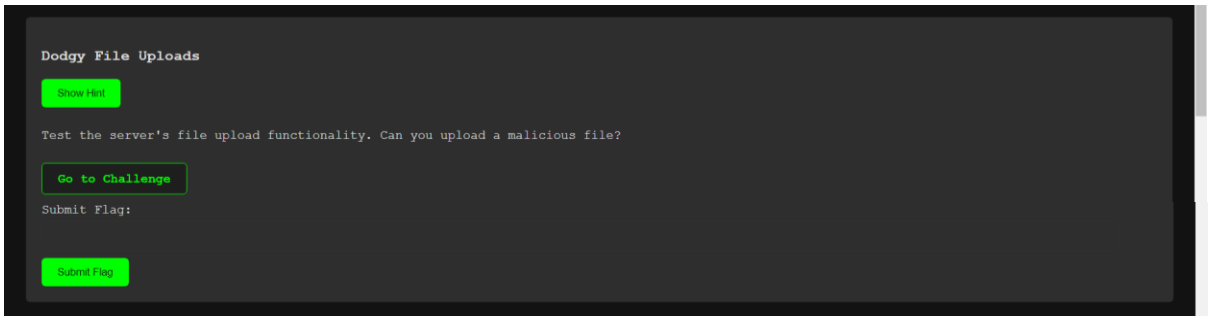
3. **Dodgy File Uploads (File Upload Vulnerability)**

Description:

This challenge involves exploiting a file upload vulnerability by uploading a malicious file. Participants need to upload a specially crafted file to trigger the vulnerability.

Hint:

"Not all files are safe to upload..."



Steps to Complete:

1. Go to the "Dodgy File Uploads" challenge page.

2. Upload a file named shell.php.

Example Content of shell.php:

php

<?php echo 'FLAG{FILE_UPLOAD_FLAG}'; ?>

Flag:

FLAG{FILE_UPLOAD_FLAG}

4. **Ping Service Exploit (Command Injection)**

Description:

This challenge involves exploiting a command injection vulnerability by executing arbitrary commands. Participants need to find a way to execute commands on the server.

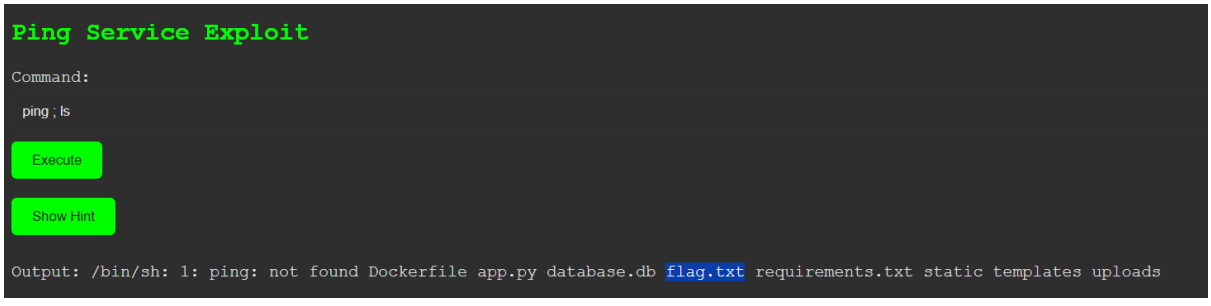
Hint:

"Sometimes commands can do more than expected..."

Steps to Complete:

1. Go to the "Ping Service Exploit" challenge page.

2. Enter commands in the input box to exploit the command injection vulnerability.

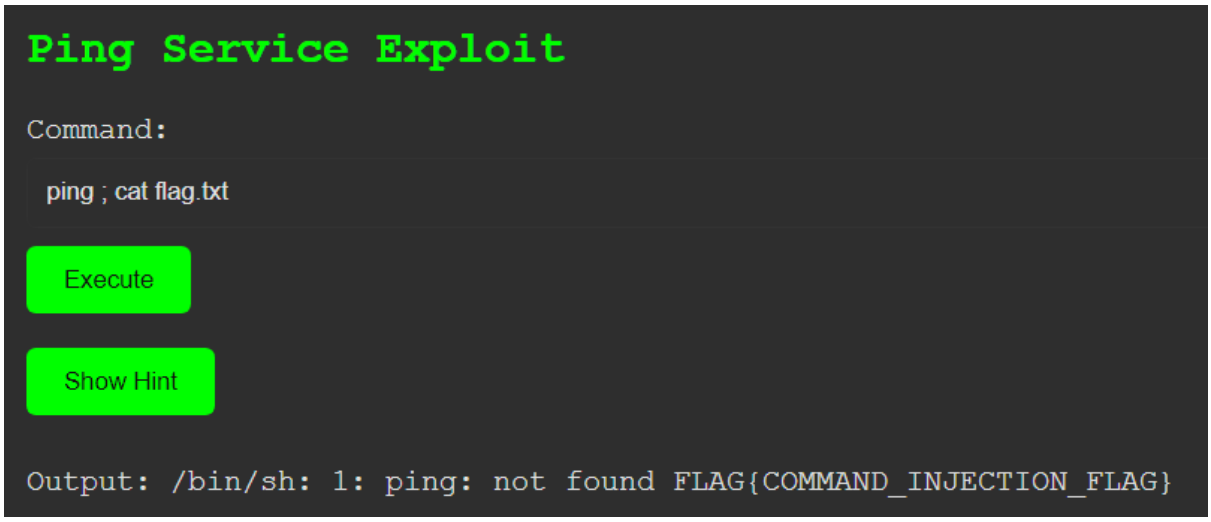


Example:

bash

ping 127.0.0.1; ls

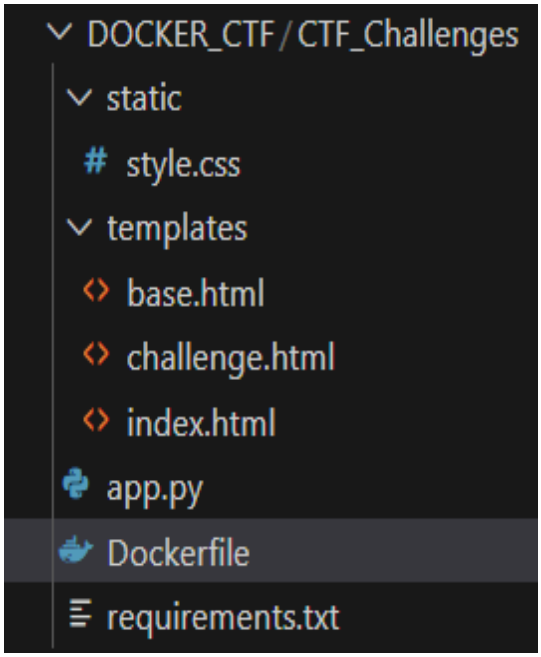
ping ; cat flag.txt



Flag:

FLAG{COMMAND_INJECTION_FLAG}

Important Files & Folder Structure



app.py

This file contains the main application code, including the setup for the challenges and routes for handling the different challenge pages.

Templates

- **base.html:** The base template for the web application.

- **index.html:** The home page template listing all challenges.

- **challenge.html:** The template used for each individual challenge.

Style

- **style.css:** The stylesheet for the web application.

Dockerfile

This file contains the instructions for building the Docker image.

requirements.txt

This file lists the Python dependencies for the application.

Conclusion

This Web Application Vulnerability Lab provides an interactive environment for learning and practicing web application security concepts. Each challenge is designed to illustrate common vulnerabilities and techniques for exploiting them. Follow the instructions provided to set up the lab and begin exploring the challenges.