**Documentation**

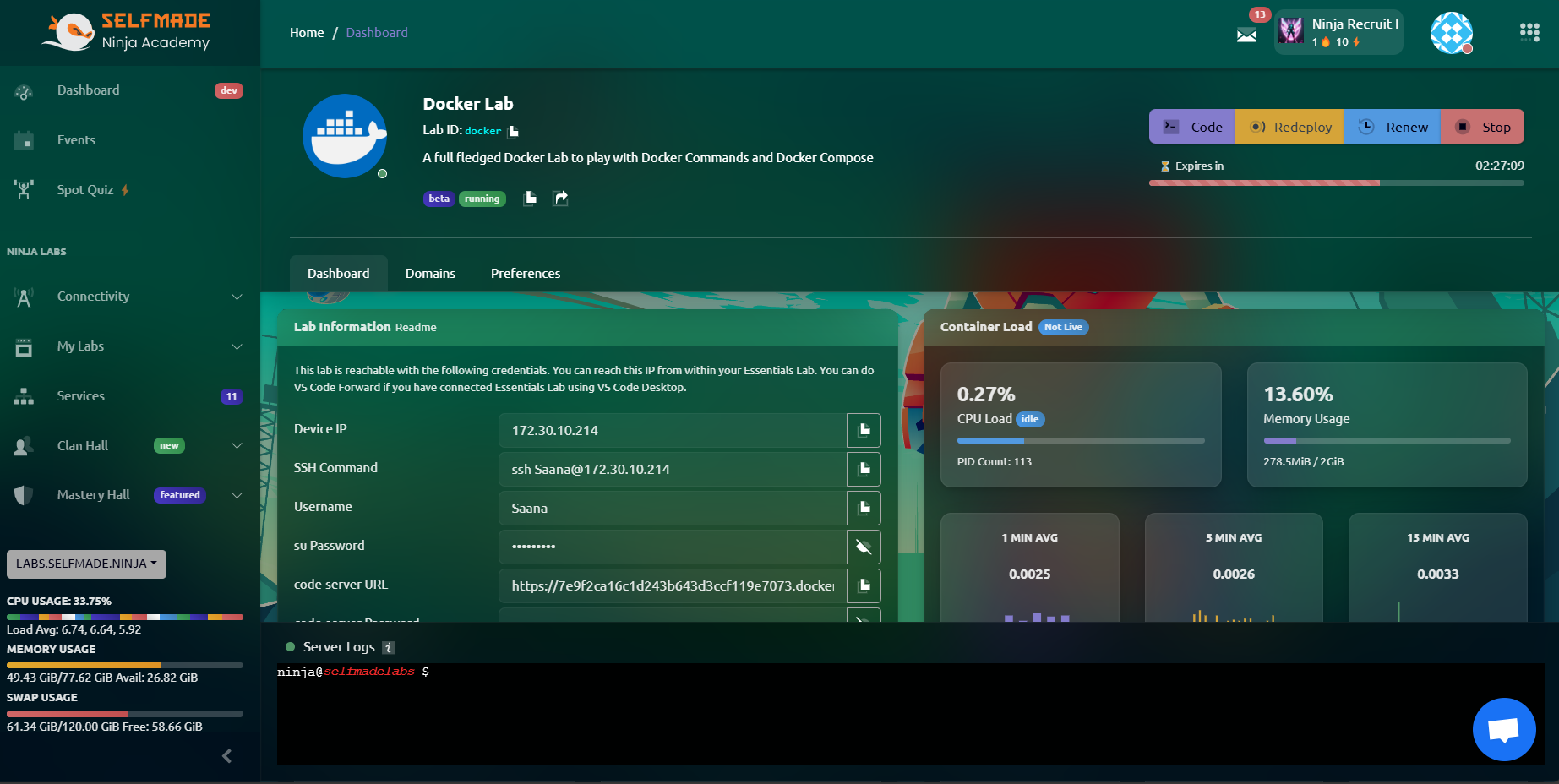
**Docker Developer Project Assignment for**

**CTF Challenge Creation**

**Creator:** Sahana M

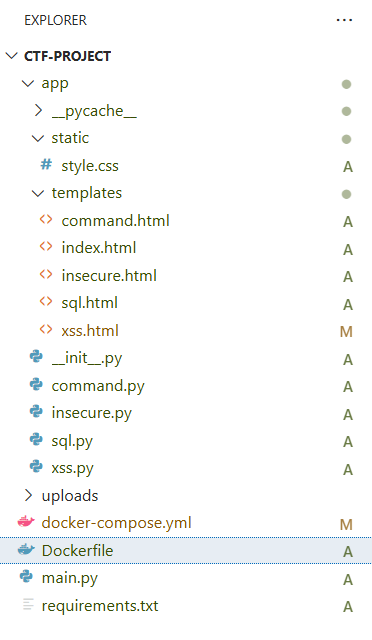
**Project Summary**

The Dockerized Web Application Vulnerability Lab is designed to help participants learn and practice web application security through a series of challenges. Each challenge involves exploiting specific vulnerabilities within a web application to discover hidden flags. This helps enhancing their understanding of various security threats and defences.  
  
**Docker Setup**   
I’m a SNA Student and I have used SNA’s Docker Lab to build this Challenge



**Directory Layout**  
Folder: CTF-Project  
This folder consists of all the necessary files required to run the CTF Challenges

**Structure of the folder:**



**Docker Commands**

# Navigate to the directory containing the Dockerfile

cd CTF-Project

# Build the Docker image

docker build -t image .

# Run the Docker container on port 7070

docker run -d -p 7070:7070 --name challenge image

**Challenges**

* SQL Injection
* XSS Injection
* Command Injection
* Insecure File Upload

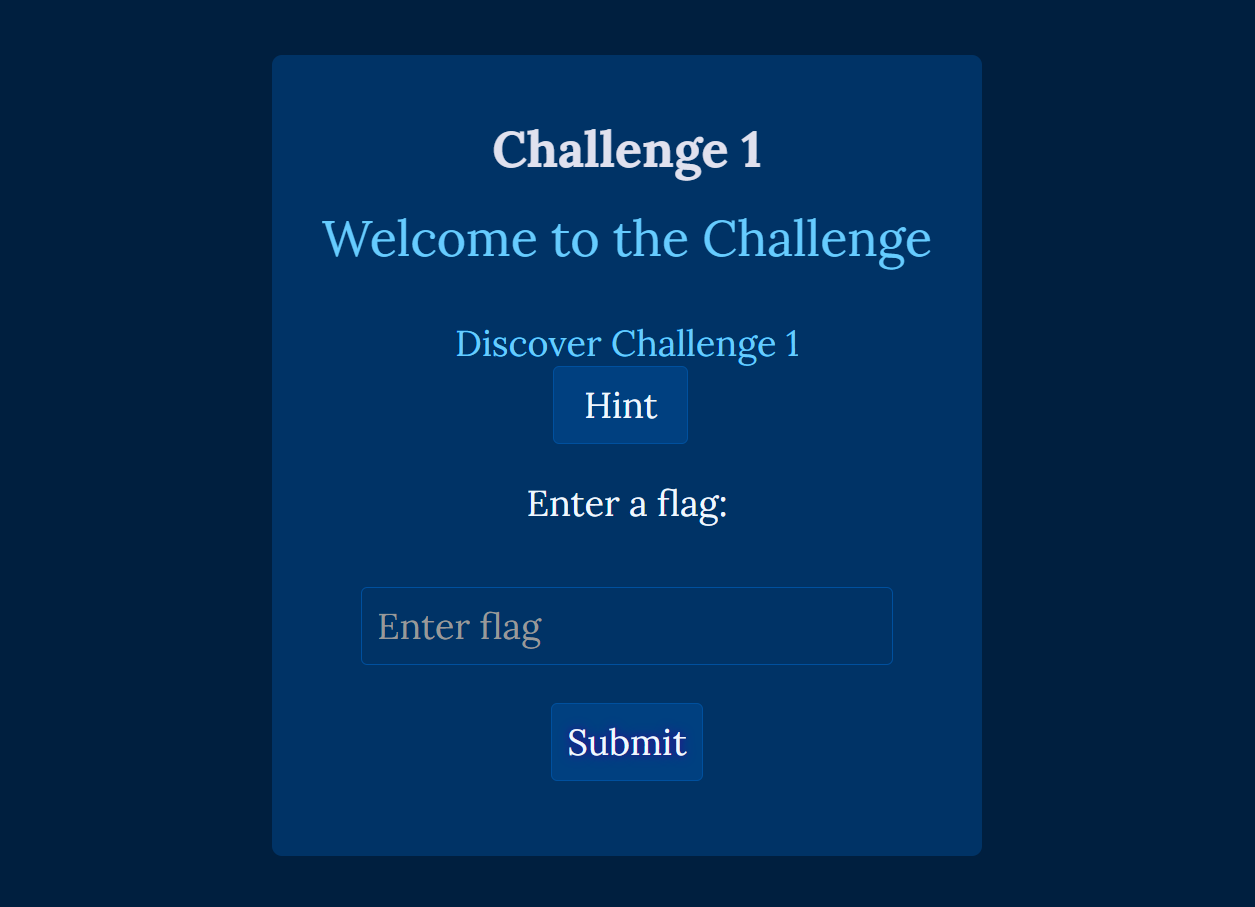
**Home Page**



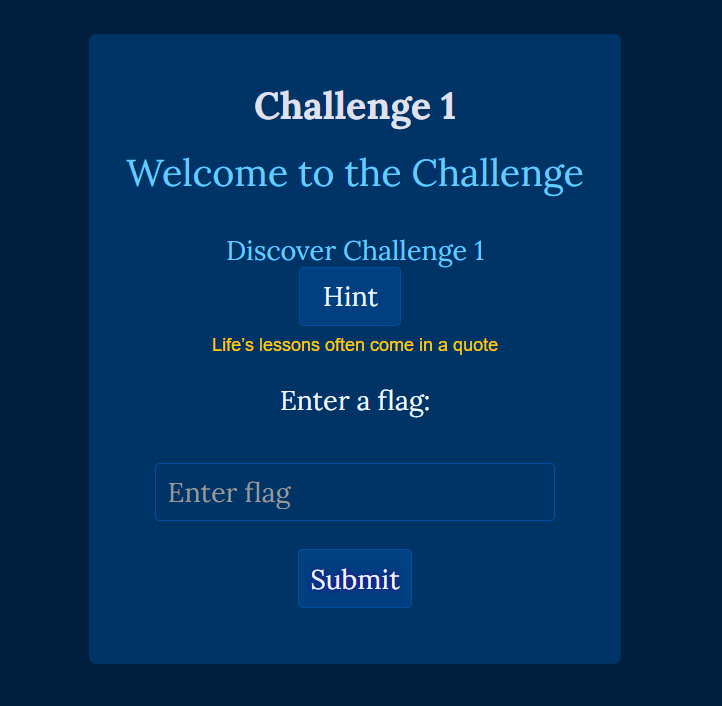
**Challenge 1 – SQL Injection**

An input form that allows participants to exploit SQL queries to retrieve hidden flags from the database.

* Navigate to “Challenge 1” page

****

* Click on the Hint Button to view the clue



**Challenge Clue:**

"Life's lessons often come in a quote"

Participants need to interpret this proverb and recognize that "quote" hints at quotation marks (' or "), suggesting that an SQL injection might be possible.

**Enter SQL Injection Payload:**

Username: Possibilities - [

" ' or 1=1--",

" ' or 1=1#",

" ' or 1=1/\*",

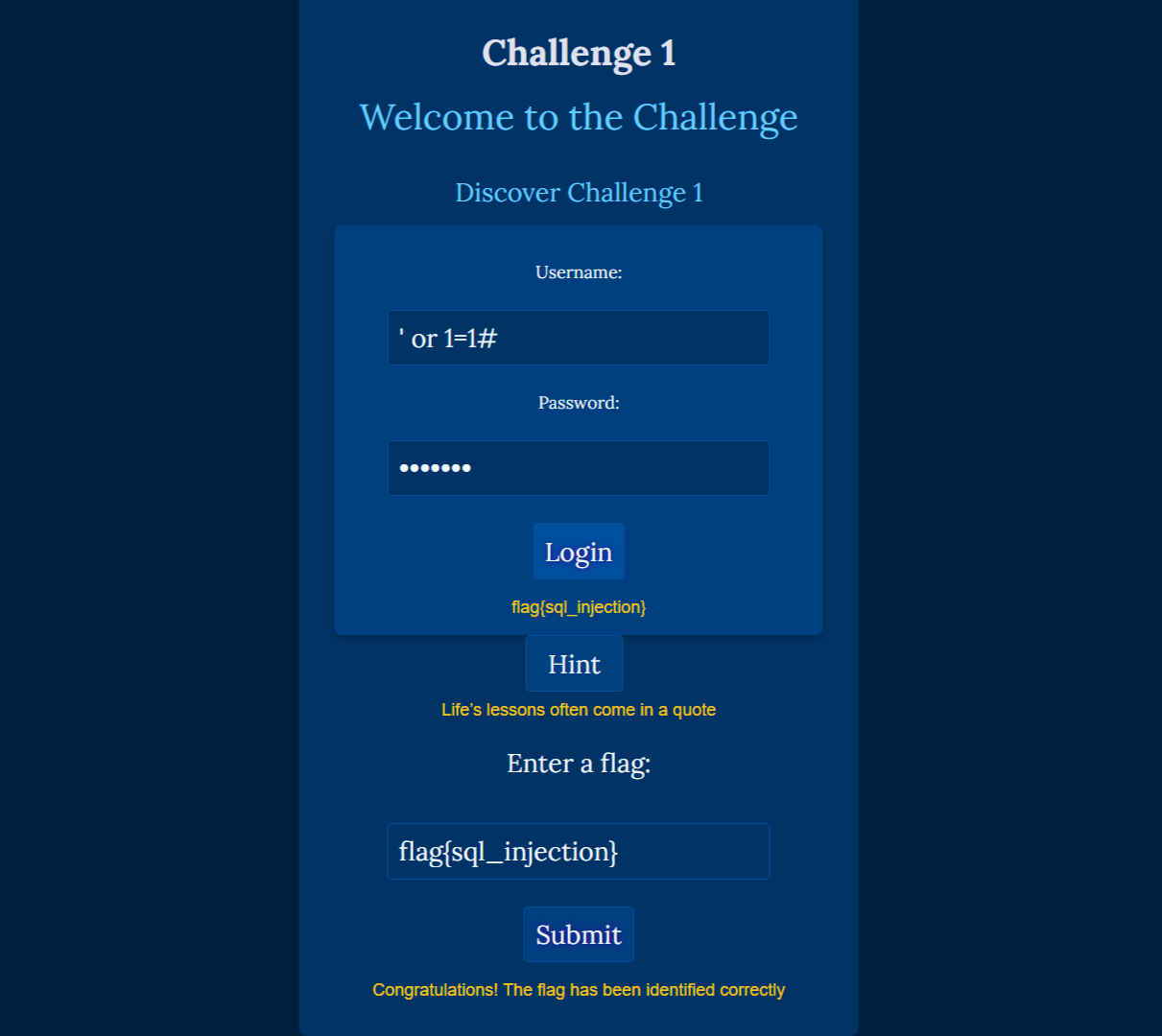
" ') or '1'='1--",

" ') or ('1'='1--"

]

Password: anything

* Press login and receive the flag
* Submit the flag and gain points



**Challenge 2 – XSS Injection**

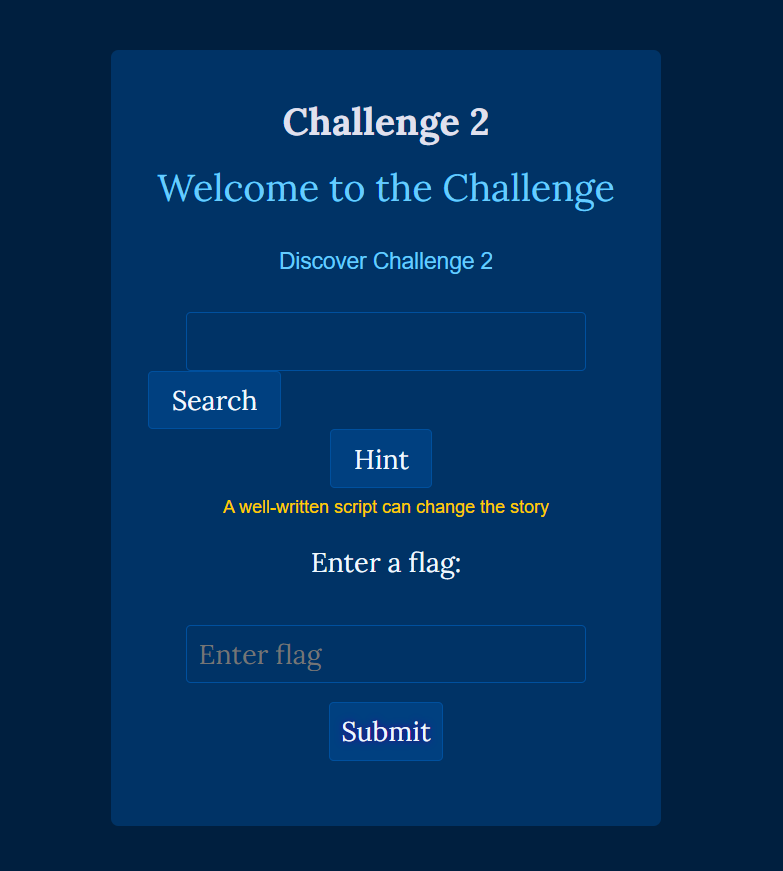
Pages where user inputs are not sanitized, allowing participants to inject scripts to uncover flags.

* Navigate to “Challenge 2” page
* Click on the Hint Button to view the clue

**Challenge Clue:**

“A well-written script can change the story”

Participants need to intercept this proverb and recognize that “script” is the clue suggesting that an XSS Injection might be possible.



**Enter XSS Injection Payload**   
  
Input: <script>alert('Hello');</script>

* Search and obtain the flag
* Submit the flag to gain points



**Challenge 3 – Command Injection**

Execute system commands based on user input, where participants can inject malicious commands.

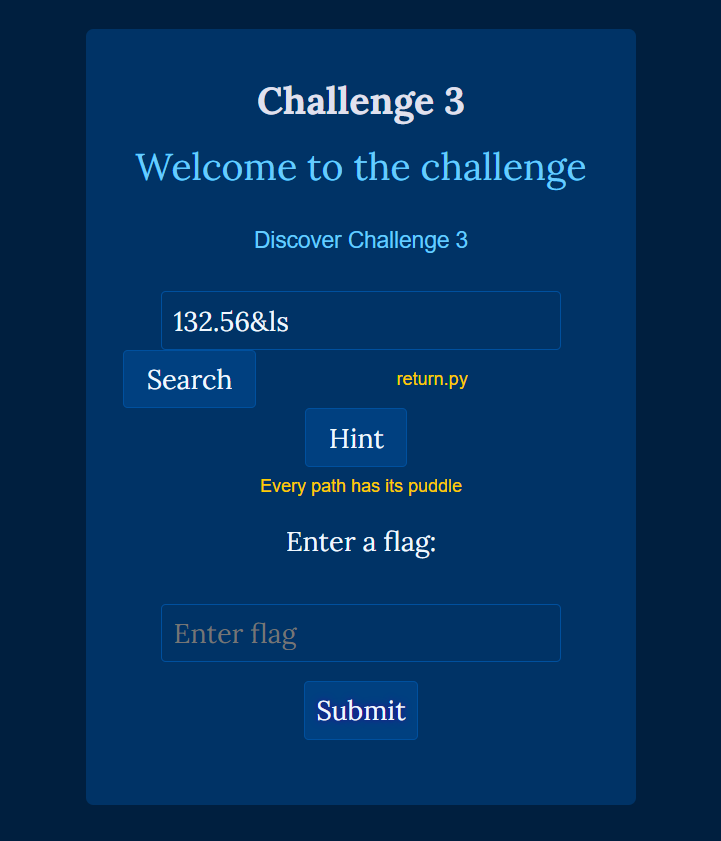
* Navigate to “Challenge 3” page
* Click on the Hint Button to view the clue

**Challenge Clue:**

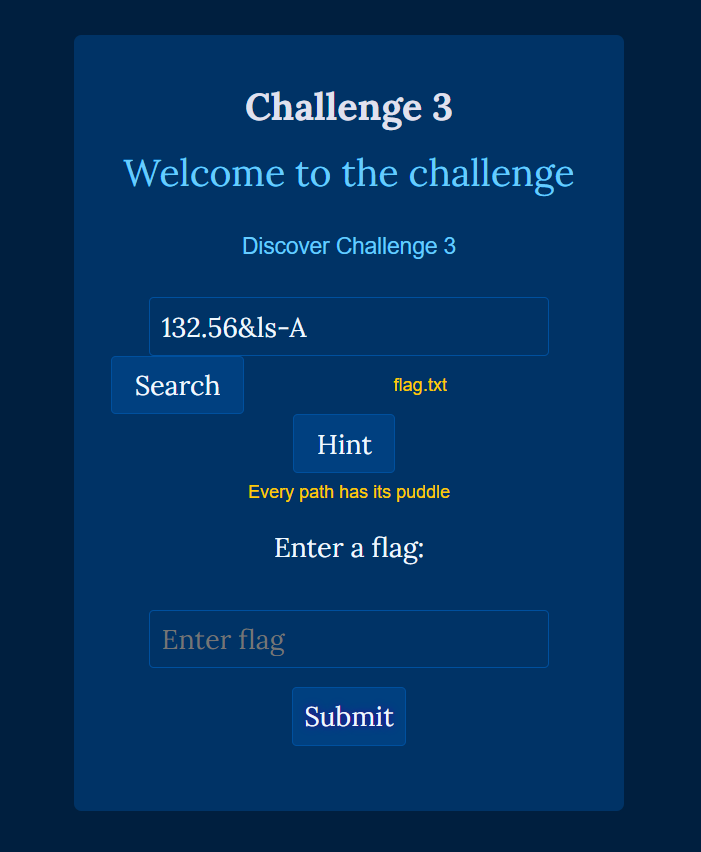
“Every path has its puddle”

Participants need to intercept this proverb and recognize that “path” hints at the command line path, suggesting that a Command Injection might be possible.

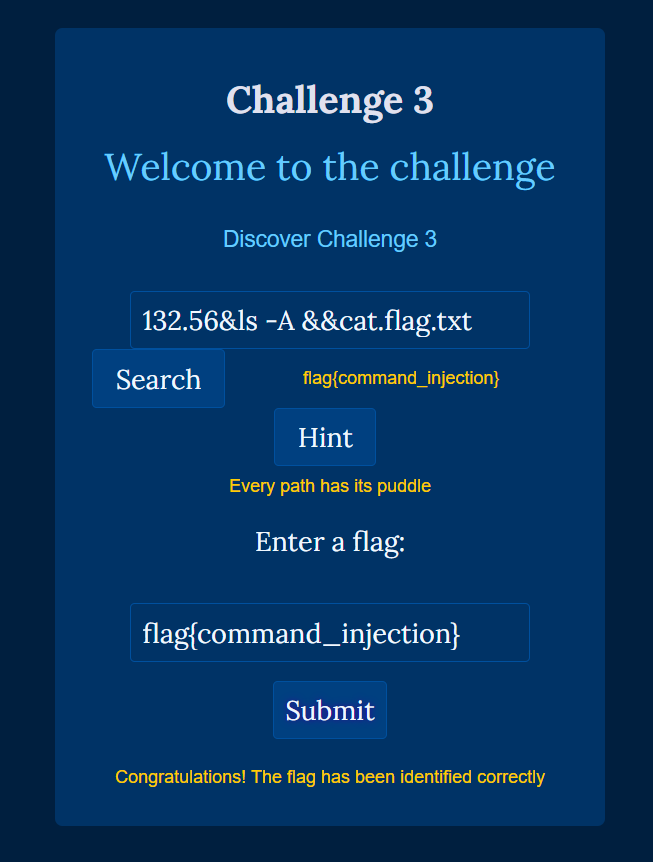
* Adding &ls to acquire the file



* To list all the files hidden inside the directory we use -A



* To read the file we use &&cat+file name

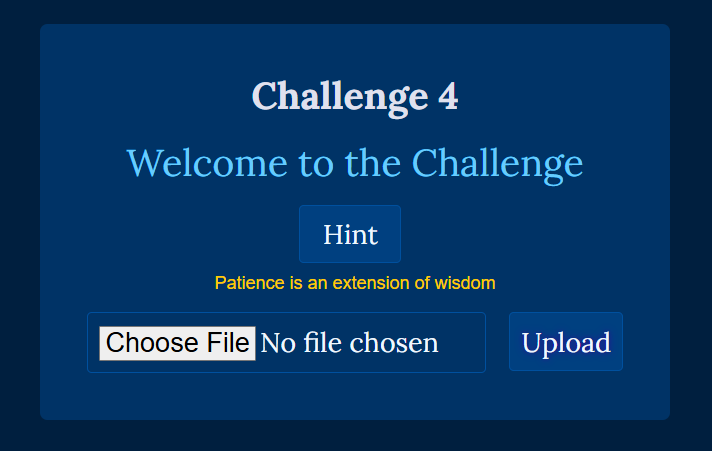


* Search and obtain the flag
* Submit the flag to gain points

**Challenge 4 – Insecure File Upload**

File upload functionality that does not properly validate or sanitize file contents, leading to potential code execution or information disclosure.

* Navigate to “Challenge 4” page
* Click on the Hint Button to view the clue



**Challenge Clue:**

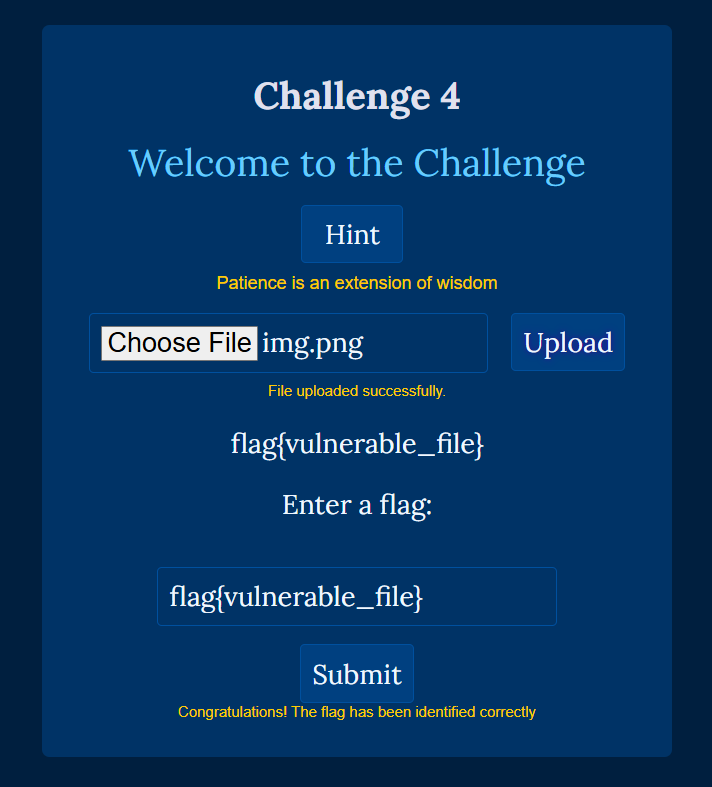
“Patience is an extension of wisdom”

Participants need to intercept this proverb and recognize that “extension” hints at the end of a file, suggesting that Insecure File Upload might be possible.

**Input Possibilities:**

{'.sh', '.txt', '.jpg', '.png'}

* Choose a file with any of these extensions
* Upload the file and obtain the flag
* Submit the flag to gain points



**Conclusion**

This CTF challenge website offers a valuable platform for cybersecurity enthusiasts to engage with and explore fundamental web application vulnerabilities in a practical manner. Through a series of carefully crafted challenges, participants are invited to identify and exploit specific weaknesses such as SQL Injection, Cross-Site Scripting (XSS), Command Injection and Insecure File Upload.