# Penetration Testing Report

**Full Name: Ilyas Bajji**
**Program: HCPT**
**Date:21/02/2024**

## Introduction

This report document hereby describes the proceedings and results of a Black Box security assessment conducted against the **Week 1 Labs**. The report hereby lists the findings and corresponding best practice mitigation actions and recommendations.

## 1. Objective

The objective of the assessment was to uncover vulnerabilities in the **Week 1 Labs** and provide a final security assessment report comprising vulnerabilities, remediation strategy and recommendation guidelines to help mitigate the identified vulnerabilities and risks during the activity.

## 2. Scope

This section defines the scope and boundaries of the project.

| | |
|---|---|
| **Application Name** | **HTML injection labs, Clickjacking labs** |

## 3. Summary

Outlined is a Black Box Application Security assessment for the **Week 1 Labs**.

**Total number of Sub-labs: 8 Sub-labs**

| High | Medium | Low |
|---|---|---|
| 4 | 3 | 1 |

| | | |
|---|---|---|
| High | - | **Number of Sub-labs with hard difficulty level** |
| Medium | - | **Number of Sub-labs with Medium difficulty level** |

Low - Number of Sub-labs with Easy difficulty level

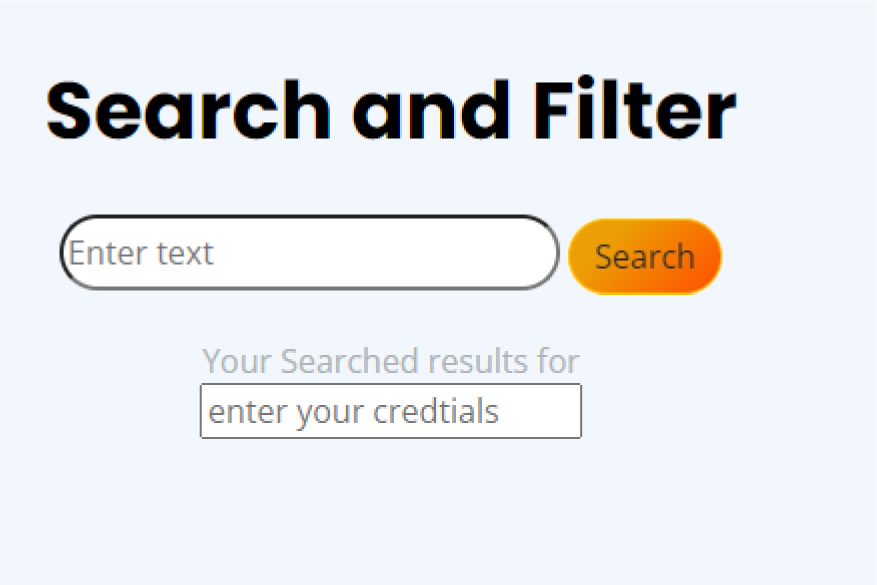
## 1. HTML injection

### 1.1. HTML’s are easy !

| Reference | Risk Rating |
|---|---|
| HTML’s are easy | **Low** |
| **Tools Used** | |
| HTML code | |
| **Vulnerability Description** | |
| Injecting html code can be executed into the page web | |
| **How It Was Discovered** | |
| Manual Analysis : <form> <input type="text" name="credentials" placeholder="enter your credtials"" ></form> | |
| **Vulnerable URLs** | |
| https://labs.hacktify.in/HTML/html_lab/lab_1/html_injection_1.php | |
| **Consequences of not Fixing the Issue** | |
| The injected code can alter the appearance,behavior,or functionality of the web page,leading to various security risks such as phishing attacks,session hijacking,or data theft. | |
| **Suggested Countermeasures** | |
| Input Validation, Output Encoding, CSP (Content Security Policy ), Parameterized Queries (Prepared Statements) | |
| **References** | |
| https://nvd.nist.gov/vuln/detail/CVE-2022-3245 | |

## Proof of Concept

section contains the proof of the above vulnerabilities as the screenshot of the vulnerability of the lab



Search and Filter

Enter text

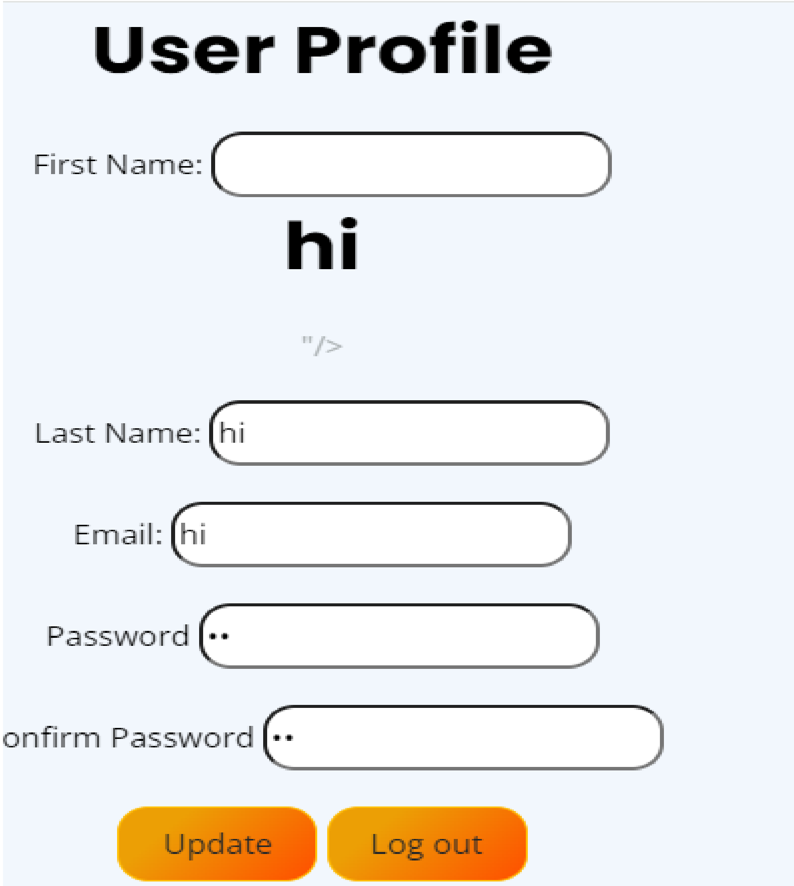
Your Searched results for

enter your credtials

## 1.2. Let me store them !

| | |
|---|---|
| **Reference** | **Risk Rating** |
| Let me store them! | **Low** |
| **Tools Used** | |
| HTML code | |
| **Vulnerability Description** | |
| Injecting html code in register mode, can be executed into the page web | |
| **How It Was Discovered** | |
| Manual Analysis : “><h1>hi</h1> | |
| **Vulnerable URLs** | |
| https://labs.hacktify.in/HTML/html_lab/lab_2/html_injection_2.php | |
| **Consequences of not Fixing the Issue** | |
| The injected code can alter the appearance,behavior,or functionality of the web page,leading to various security risks such as phishing attacks,session hijacking,or data theft. | |
| **Suggested Countermeasures** | |
| Input Validation, Output Encoding, CSP (Content Security Policy ), Parameterized Queries (Prepared Statements) | |
| **References** | |
| https://nvd.nist.gov/vuln/detail/CVE-2022-3245 | |

## Proof of Concept



User Profile

First Name:

hi

"/>

Last Name: hi

Email: hi

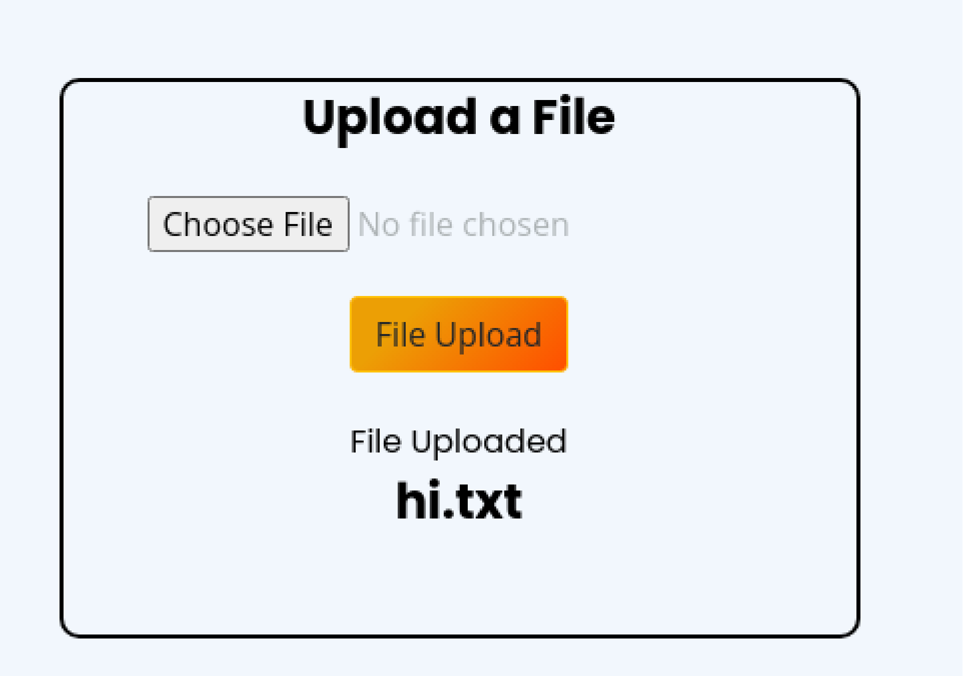
Password ••

onfirm Password ••

## 1.3. File Names are also vulnerable !

| Reference | Risk Rating |
|---|---|
| File Names are also vulnerable ! | low |
| **Tools Used** | |
| HTML code , burp suite | |
| **Vulnerability Description** | |
| Injecting html code in file name can be executed into the page web | |
| **How It Was Discovered** | |
| Manual Analysis : <h1>hi.txt | |
| **Vulnerable URLs** | |
| https://labs.hacktify.in/HTML/html_lab/lab_3/html_injection_3.php | |
| **Consequences of not Fixing the Issue** | |
| The injected code can alter the appearance,behavior,or functionality of the web page,leading to various security risks such as phishing attacks,session hijacking,or data theft. | |
| **Suggested Countermeasures** | |
| Input Validation, Output Encoding, CSP (Content Security Policy ), Parameterized Queries (Prepared Statements) | |
| **References** | |
| https://nvd.nist.gov/vuln/detail/CVE-2022-3245 | |

## Proof of Concept


Upload a File

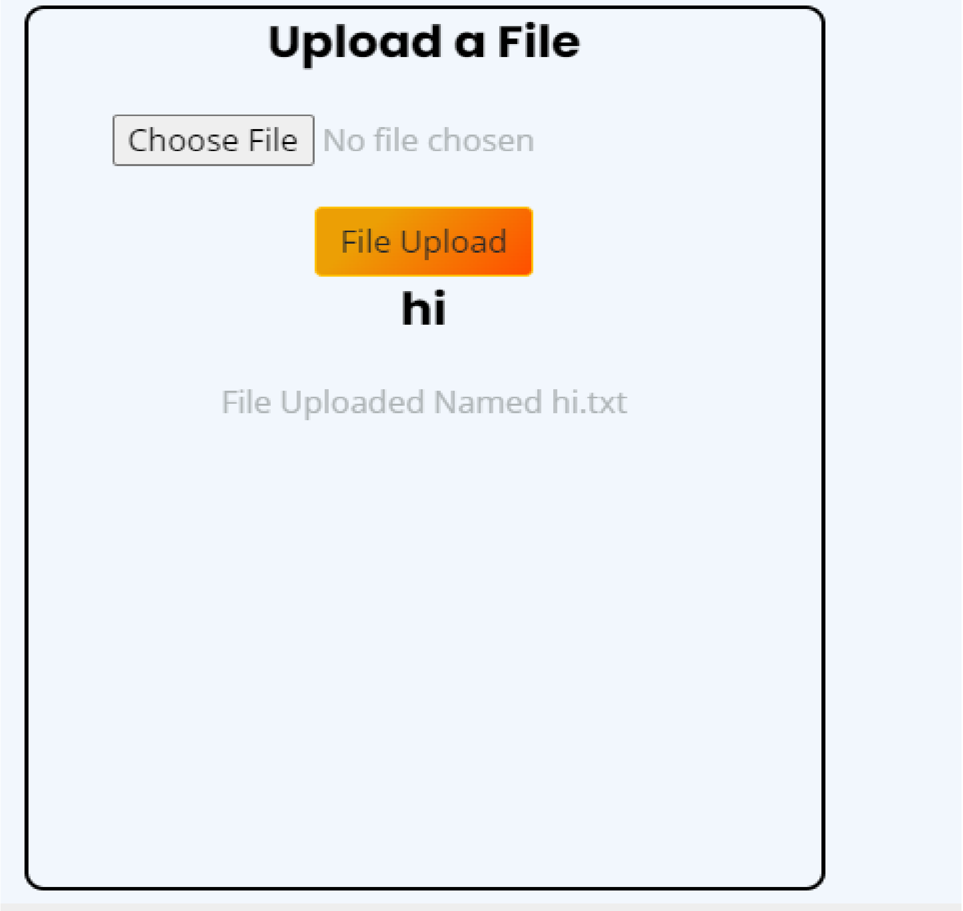
File Uploaded

hi.txt

## 1.4 File Content And HTML Injection A Perfect Pair !

| Reference | Risk Rating |
|---|---|
| File Content And HTML Injection A Perfect Pair | **Medium** |
| **Tools Used** | |
| HTML code | |
| **Vulnerability Description** | |
| Vulnerable Field: File Content Parameter | |
| **How It Was Discovered** | |
| Manual Analysis : add to file content : <h1>hi</hi> | |
| **Vulnerable URLs** | |
| https://labs.hacktify.in/HTML/html_lab/lab_4/html_injection_4.php | |
| **Consequences of not Fixing the Issue** | |
| The injected code can alter the appearance,behavior,or functionality of the web page,leading to various security risks such as phishing attacks,session hijacking,or data theft.. or Execute a shell code. | |
| **Suggested Countermeasures** | |
| Input Validation, Output Encoding, CSP (Content Security Policy ), Parameterized Queries (Prepared Statements) | |
| **References** | |
| https://nvd.nist.gov/vuln/detail/CVE-2022-3245 | |

## Proof of Concept



Upload a File

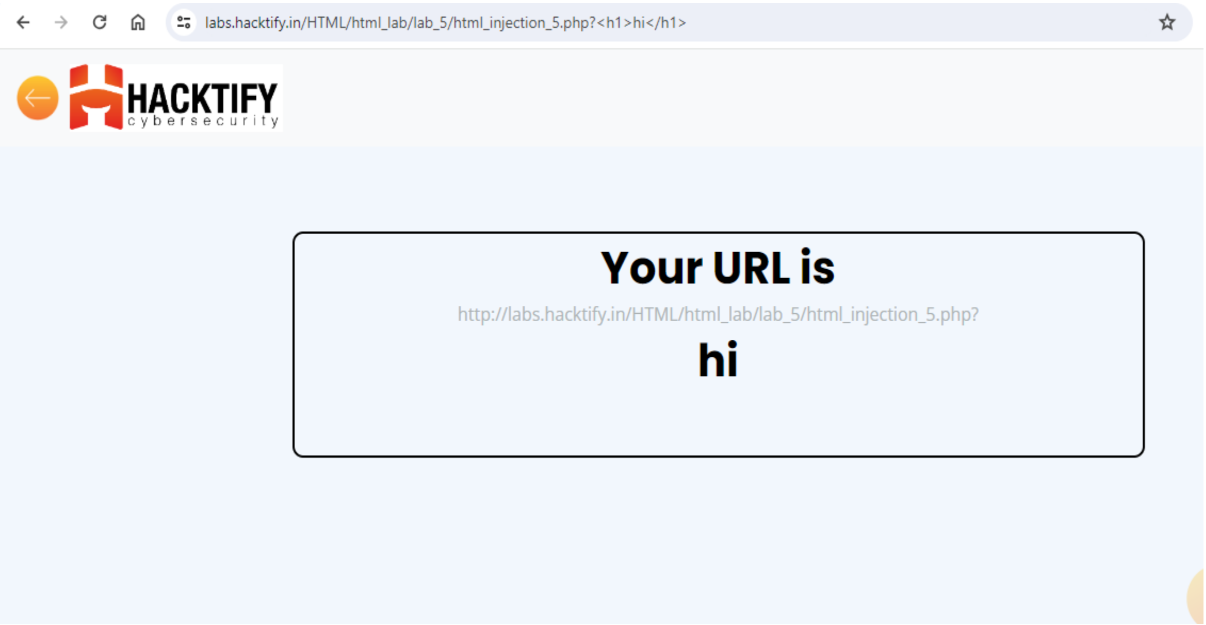
hi

File Uploaded Named hi.txt

## 1.5. Injecting HTML using URL

| Reference | Risk Rating |
|---|---|
| Injecting HTML using URL | **Medium** |
| **Tools Used** | |
| HTML code | |
| **Vulnerability Description** | |
| Injecting html code can be executed into the page web | |
| **How It Was Discovered** | |
| Manual Analysis : inject in URL : ?<h1>hi</h1> | |
| **Vulnerable URLs** | |
| https://labs.hacktify.in/HTML/html_lab/lab_5/html_injection_5.php | |
| **Consequences of not Fixing the Issue** | |
| The injected code can alter the appearance,behavior,or functionality of the web page,leading to various security risks such as phishing attacks,session hijacking,or data theft. | |
| **Suggested Countermeasures** | |
| Input Validation, Output Encoding, CSP (Content Security Policy ), Parameterized Queries (Prepared Statements) | |
| **References** | |
| https://nvd.nist.gov/vuln/detail/CVE-2022-3245 | |

## Proof of Concept



labs.hacktify.in/HTML/html_lab/lab_5/html_injection_5.php?<h1>hi</h1>

Your URL is

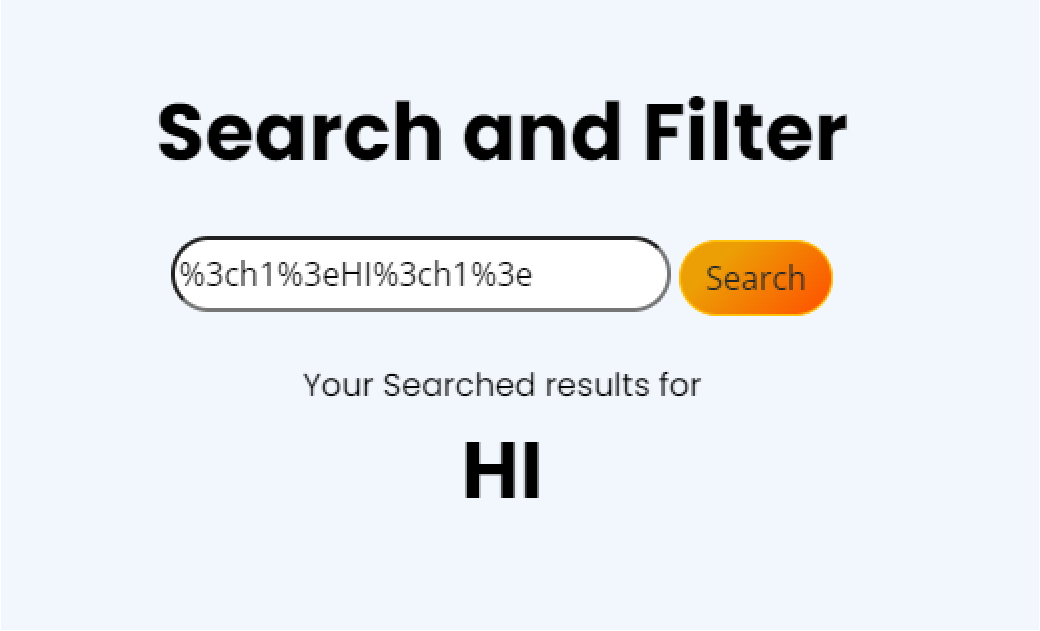
http://labs.hacktify.in/HTML/html_lab/lab_5/html_injection_5.php?

hi

## 1.6. Encode it!

| Reference | Risk Rating |
|---|---|
| Encode it ! | **Low** |
| **Tools Used** | |
| HTML code | |
| **Vulnerability Description** | |
| Injecting html code can be executed into the page web | |
| **How It Was Discovered** | |
| Manual Analysis : by using url encoding %3ch1%3eHI%3ch1%3e | |
| **Vulnerable URLs** | |
| https://labs.hacktify.in/HTML/html_lab/lab_6/html_injection_6.php | |
| **Consequences of not Fixing the Issue** | |
| The injected code can alter the appearance,behavior,or functionality of the web page,leading to various security risks such as phishing attacks,session hijacking,or data theft. | |
| **Suggested Countermeasures** | |
| Input Validation, Output Encoding, CSP (Content Security Policy ), Parameterized Queries (Prepared Statements) | |
| **References** | |
| https://nvd.nist.gov/vuln/detail/CVE-2022-3245 | |

## Proof of Concept



Search and Filter

%3ch1%3eHI%3ch1%3e

Your Searched results for

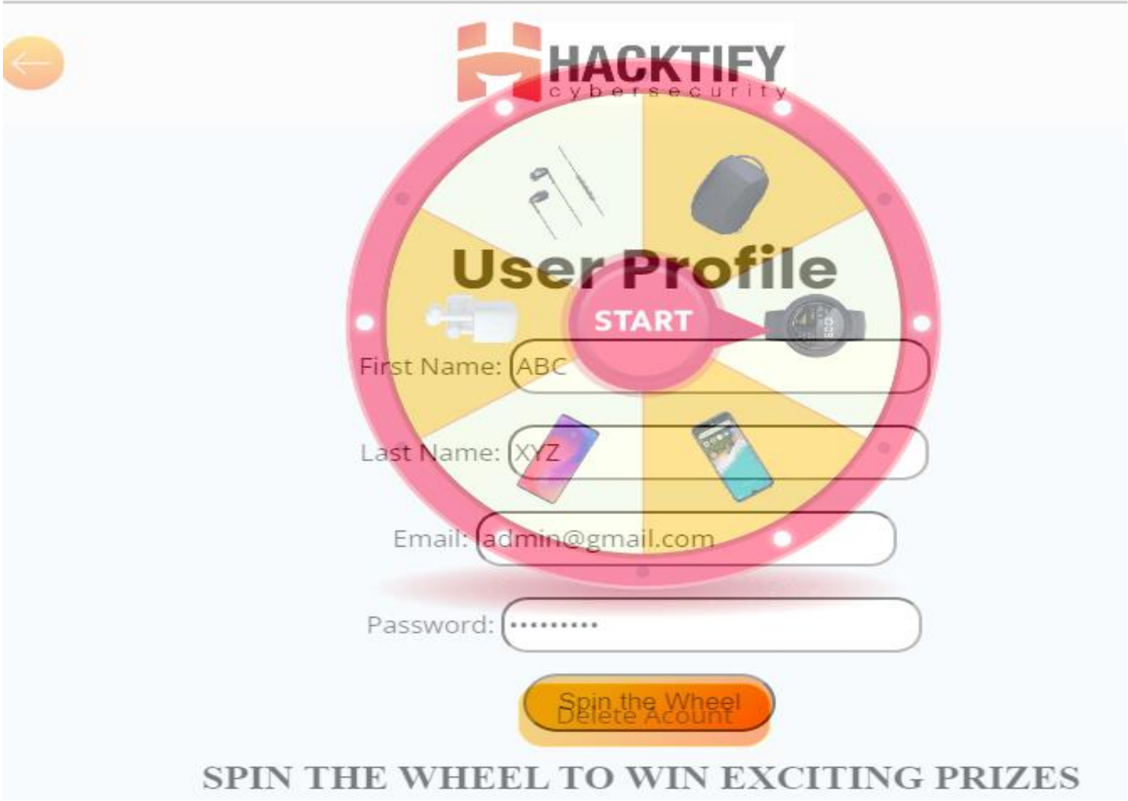
HI

## 2. Clickjacking

### 2.1 Let’s hijack

| Reference | Risk Rating |
|---|---|
| Let’s hijack | Low |
| **Tools Used** | |
| observation | |
| **Vulnerability Description** | |
| Hide tricky buttons or process over a legitimate content i.e button of spin wheel that it delets an accounts instead . | |
| **How It Was Discovered** | |
| Click test button | |
| **Vulnerable URLs** | |
| https://labs.hacktify.in/HTML/clickjacking_lab/lab_1/lab_1.php | |
| **Consequences of not Fixing the Issue** | |
| exploit the inherent trust users place in familiar websites and interfaces to deceive them into unwittingly executing malicious actions. | |
| **Suggested Countermeasures** | |
| X-Frame-Options Header, CSP (Content Security Policy), Frame Busting Javascript Code | |
| **References** | |
| https://portswigger.net/web-security/clickjacking | |

## Proof of Concept



User Profile

First Name: ABC

Last Name: XYZ

Email: admin@gmail.com

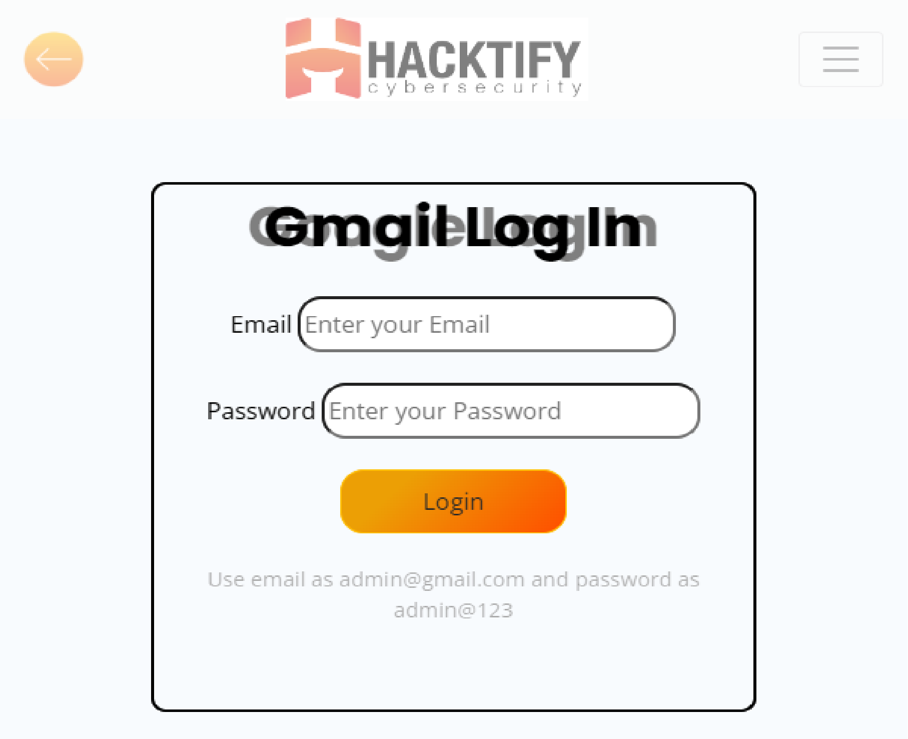
Password: ••••••••••

SPIN THE WHEEL TO WIN EXCITING PRIZES

## 2.2. Re-Hijack!

| **Reference** | **Risk Rating** |
|---|---|
| Re-Hijack! | **Medium** |
| **Tools Used** | |
| Tools that you have used to find the vulnerability. | |
| **Vulnerability Description** | |
| Trick name : using gmail instead of google even they are the same ; gmail credentials give access to all google . | |
| **How It Was Discovered** | |
| Manual Analysis: click TEST button | |
| **Vulnerable URLs** | |
| https://labs.hacktify.in/HTML/clickjacking_lab/lab_2/testclickjacking.php | |
| **Consequences of not Fixing the Issue** | |
| exploit the inherent trust users place in familiar websites and interfaces to deceive them into unwittingly executing malicious actions. | |
| **Suggested Countermeasures** | |
| X-Frame-Options Header, CSP (Content Security Policy), Frame Busting Javascript Code | |
| **References** | |
| https://portswigger.net/web-security/clickjacking | |

## Proof of Concept



GmgilLogIn

Email Enter your Email

Password Enter your Password

Use email as admin@gmail.com and password as admin@123