Week 2
Penetration Testing Report

Introduction

This report document hereby describes the proceedings and results of a Black Box security assessment conducted against the **Week 2 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 2 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 and Clickjacking**

3. Summary

Outlined is a Black Box Application Security assessment for the **Week 2 Labs**.

Total number of Sub-labs: 8 Sub-labs

High	Medium	Low
1	3	4

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**

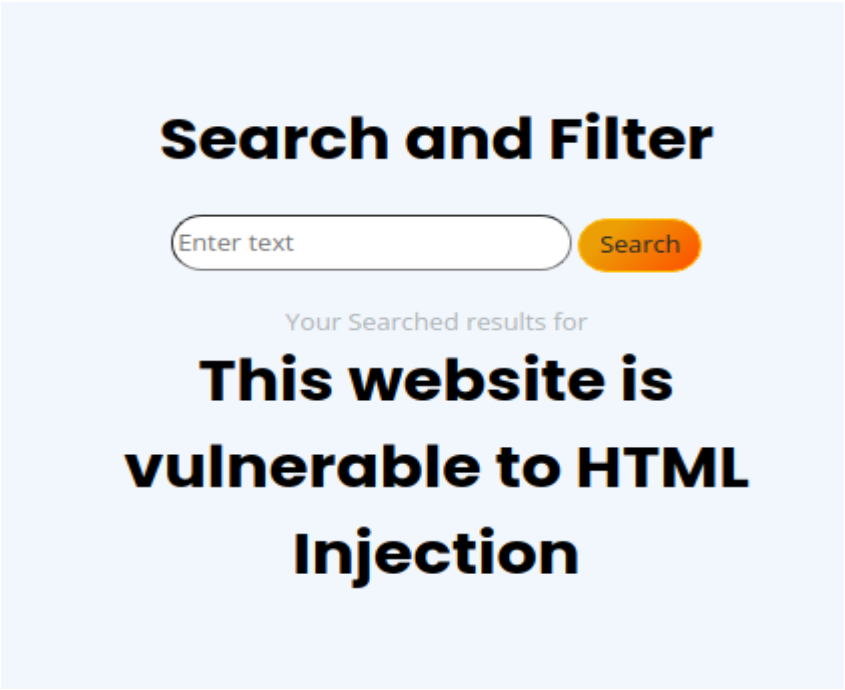
1. HTML Injection

1.1. HTML's Are Easy!

Reference	Risk Rating
HTML's Are Easy!	**Low**
Tools Used	
Browser	
Vulnerability Description	
The vulnerability is HTML Injection that allows users to execute HTML codes in the input fields.	
How It Was Discovered	
Manual Analysis - Pass any HTML code in the input field and it will get executed.	
Vulnerable URLs	
https://www.bugbountyhunter.org/internship_labs/HTML/html_lab/lab_1/html_injection_1.php	
Consequences of not Fixing the Issue	
The attacker can perform any action on the web page and can also create it as a phishing page to divert all users to other attacker-controlled web pages.	
Suggested Countermeasures	
Every input should be checked and validated.	
References	
https://www.softwaretestinghelp.com/html-injection-tutorial/	

Proof of Concept

The proof of the above vulnerability.

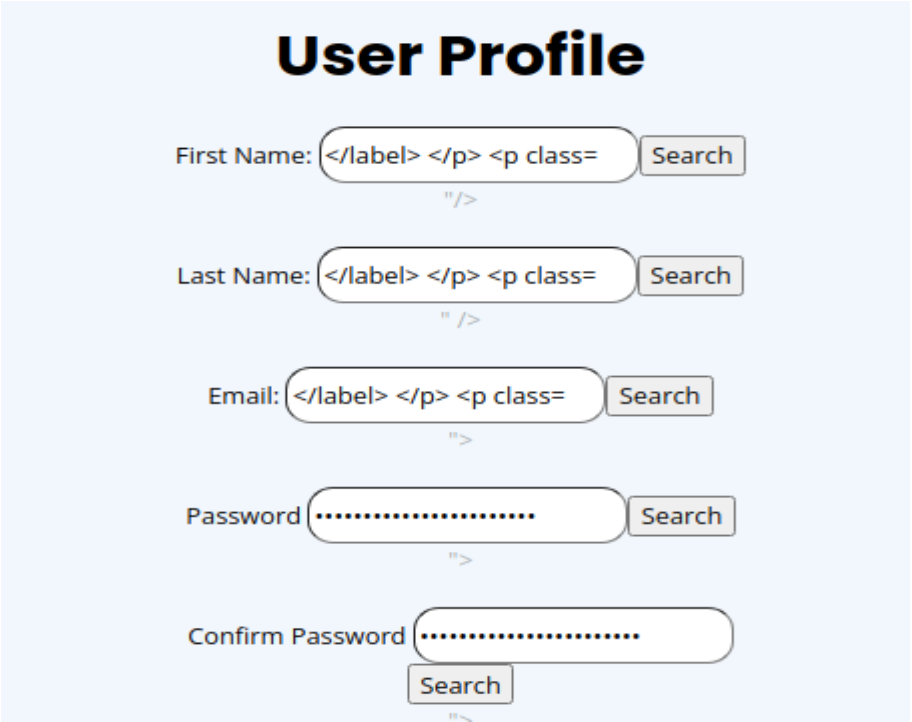


1.2. Let Me Store Them!

Reference	Risk Rating
Let Me Store Them!	**Low**
Tools Used	
Browser	
Vulnerability Description	
The vulnerability is HTML Injection that allows users to execute HTML codes in the input fields.	
How It Was Discovered	
Manual Analysis - Pass any HTML code in the input field and save the details, when we login again it will get executed.	
Vulnerable URLs	
https://www.bugbountyhunter.org/internship_labs/HTML/html_lab/lab_2/html_injection_2.php	
Consequences of not Fixing the Issue	
The attacker can perform any action on the web page and can also create it as a phishing page to divert all users to other attacker-controlled web pages.	
Suggested Countermeasures	
Every input should be checked and validated.	
References	
https://www.softwaretestinghelp.com/html-injection-tutorial/	

Proof of Concept

The proof of the above vulnerability.

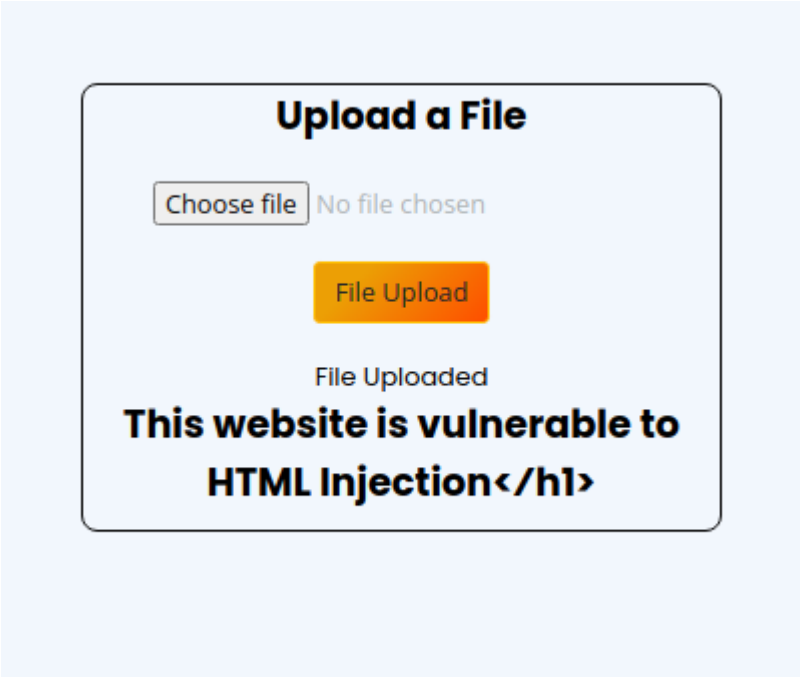


1.3. File Names Are Also Vulnerable!

Reference	Risk Rating
File Names Are Also Vulnerable!	**Low**
Tools Used	
Browser, Notepad	
Vulnerability Description	
The vulnerability is HTML Injection that allows users to execute HTML codes in the input fields.	
How It Was Discovered	
Manual Analysis - Save the file name with some HTML code, when the file is uploaded it will get executed.	
Vulnerable URLs	
https://www.bugbountyhunter.org/internship_labs/HTML/html_lab/lab_3/html_injection_3.php	
Consequences of not Fixing the Issue	
The attacker can perform any action on the web page and can also create it as a phishing page to divert all users to other attacker-controlled web pages.	
Suggested Countermeasures	
Every input should be checked and validated.	
References	
https://www.softwaretestinghelp.com/html-injection-tutorial/	

Proof of Concept

The proof of the above vulnerability.

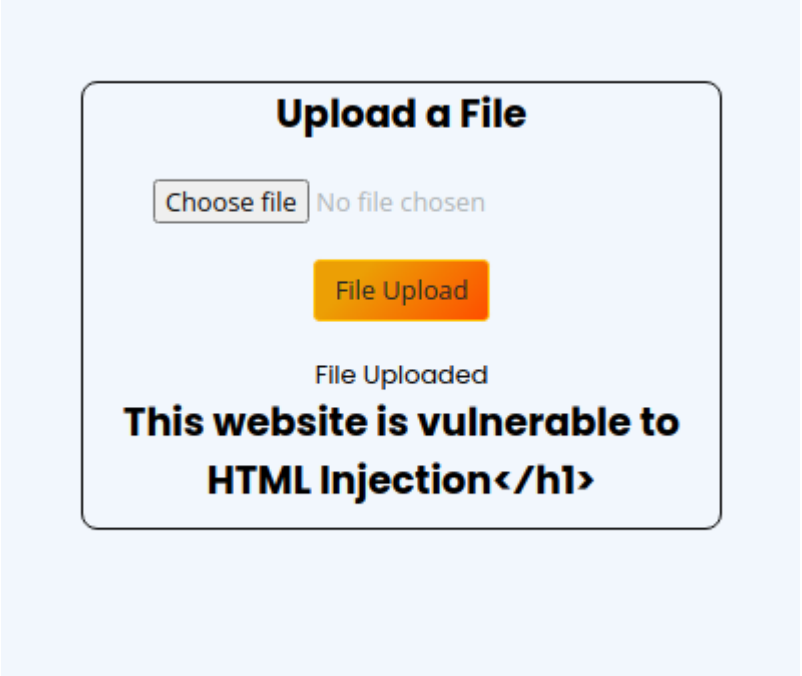


1.4. File Content And HTML Injection A Perfect Pair!

Reference	Risk Rating
File Content And HTML Injection A Perfect Pair!	**Medium**
Tools Used	
Browser, Notepad	
Vulnerability Description	
The vulnerability is HTML Injection that allows users to execute HTML codes in the input fields.	
How It Was Discovered	
Manual Analysis - Save the file name and file with some HTML code, when the file is uploaded it will get executed.	
Vulnerable URLs	
https://www.bugbountyhunter.org/internship_labs/HTML/html_lab/lab_4/html_injection_4.php	
Consequences of not Fixing the Issue	
The attacker can perform any action on the web page and can also create it as a phishing page to divert all users to other attacker-controlled web pages.	
Suggested Countermeasures	
Every input should be checked and validated.	
References	
https://www.w3schools.com/tags/att_input_type_file.asp	

Proof of Concept

The proof of the above vulnerability.



1.5. Injecting HTML Using URL!

Reference	Risk Rating
Injecting HTML Using URL!	**Medium**
Tools Used	
Browser	
Vulnerability Description	
The vulnerability is HTML Injection that allows users to execute HTML codes in the input fields.	
How It Was Discovered	
Manual Analysis - Pass some HTML code after ?id= and see it gets executed.	
Vulnerable URLs	
https://www.bugbountyhunter.org/internship_labs/HTML/html_lab/lab_5/html_injection_5.php	
Consequences of not Fixing the Issue	
The attacker can perform any action on the web page and can also create it as a phishing page to divert all users to other attacker-controlled web pages.	
Suggested Countermeasures	
Every input should be checked and validated.	
References	
https://www.softwaretestinghelp.com/html-injection-tutorial/	

Proof of Concept

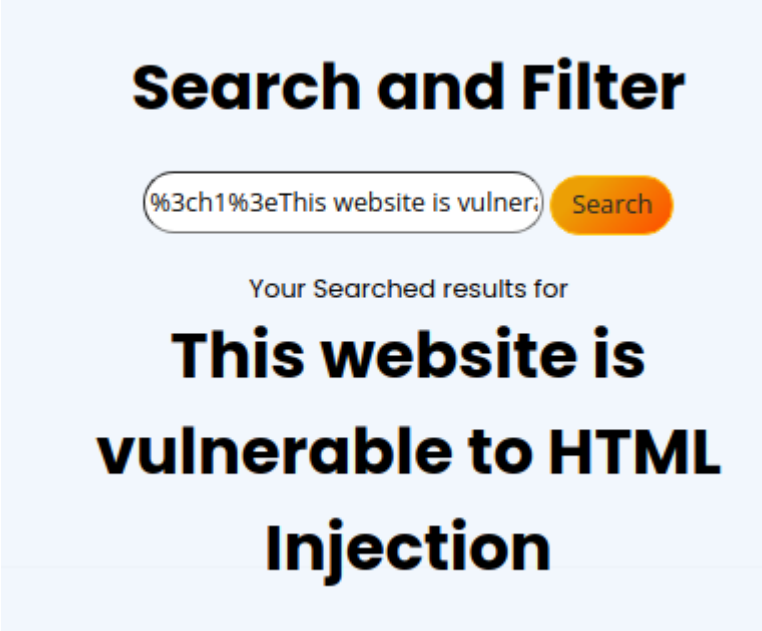
The proof of the above vulnerability.

1.6. Encode IT!

Reference	Risk Rating
Encode IT!	**High**
Tools Used	
Browser, URL Encoder	
Vulnerability Description	
The vulnerability is HTML Injection that allows users to execute HTML codes in the input fields.	
How It Was Discovered	
Manual Analysis - Pass some HTML code with encoded braces and see it gets executed.	
Vulnerable URLs	
https://www.bugbountyhunter.org/internship_labs/HTML/html_lab/lab_6/html_injection_6.php	
Consequences of not Fixing the Issue	
The attacker can perform any action on the web page and can also create it as a phishing page to divert all users to other attacker-controlled web pages.	
Suggested Countermeasures	
Every input should be checked and validated.	
References	
https://www.w3schools.com/tags/ref_urlencode.asp	

Proof of Concept

The proof of the above vulnerability.



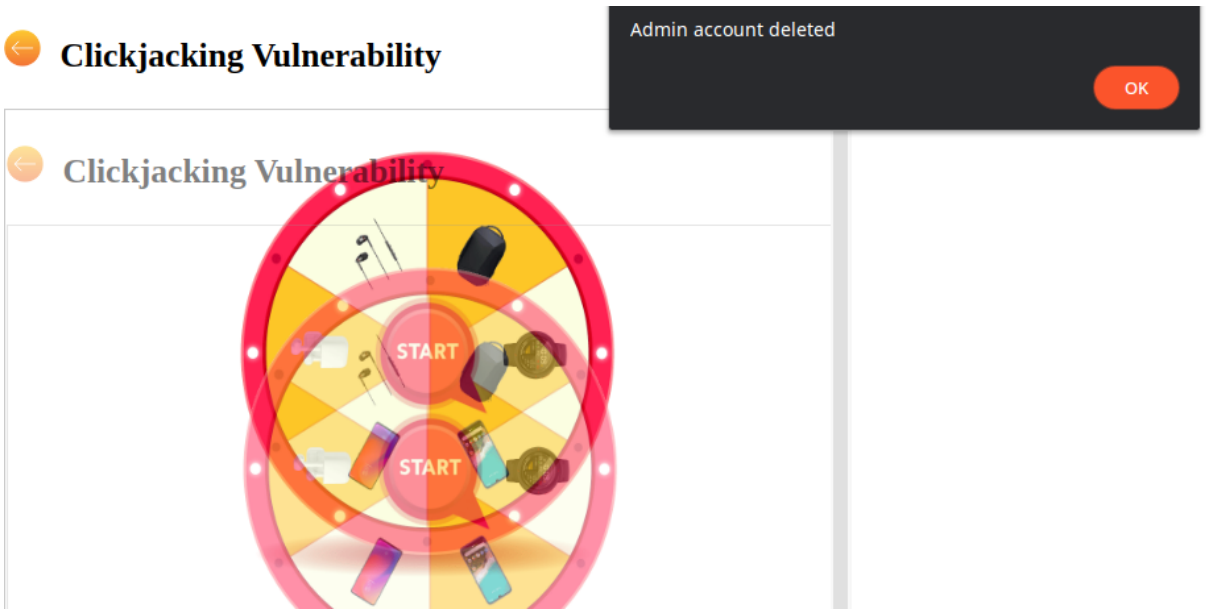
2. Clickjacking

2.1. Let's Hijack!

Reference	Risk Rating
Let's Hijack	**Low**
Tools Used	
Browser	
Vulnerability Description	
It is an interface-based attack in which a user is tricked into clicking on actionable content on a hidden website by clicking on some other content in a decoy website.	
How It Was Discovered	
Manual Analysis	
Vulnerable URLs	
https://www.bugbountyhunter.org/internship_labs/HTML/clickjacking_lab/lab_1/testclickjacking.php	
Consequences of not Fixing the Issue	
It is based on the attacker's imagination, he can make the user do anything without knowledge.	
Suggested Countermeasures	
Implement CSP	
References	
https://portswigger.net/web-security/clickjacking	

Proof of Concept

The proof of the above vulnerability.



2.2. Re-Hijack!

Reference	Risk Rating
Re-Hijack!	**Medium**
Tools Used	
Browser	
Vulnerability Description	
It is an interface-based attack in which a user is tricked into clicking on actionable content on a hidden website by clicking on some other content in a decoy website.	
How It Was Discovered	
Manual Analysis	
Vulnerable URLs	
https://www.bugbountyhunter.org/internship_labs/HTML/clickjacking_lab/lab_2/testclickjacking.php	
Consequences of not Fixing the Issue	
It is based on the attacker's imagination, he can make the user do anything without knowledge.	
Suggested Countermeasures	
Implement CSP	
References	
https://portswigger.net/web-security/clickjacking	

Proof of Concept

The proof of the above vulnerability.