Week 3
Penetration Testing Report

Introduction

This report document hereby describes the proceedings and results of a Black Box security assessment conducted against the **Week 3 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 3 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	**Cross-Site Scripting**

3. Summary

Outlined is a Black Box Application Security assessment for the **Week 3 Labs**.

Total number of Sub-labs: 11 Sub-labs

High	Medium	Low
3	3	5

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**

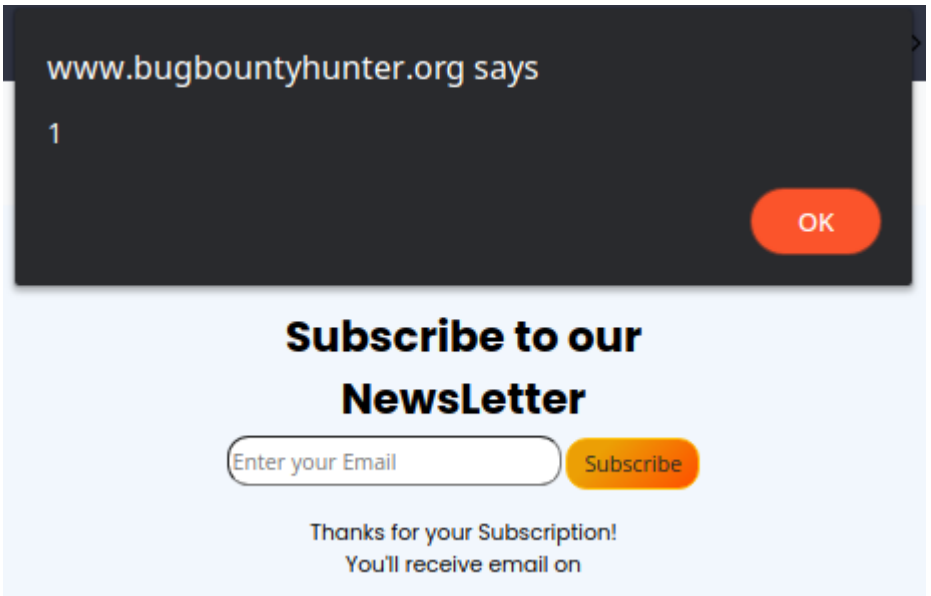
1. Cross-Site Scripting

1.1. Let's Do IT!

Reference	Risk Rating
Let's Do IT!	**Low**
Tools Used	
Browser	
Vulnerability Description	
The vulnerability is Cross-Site Scripting that allows users to execute JS codes in the input fields.	
How It Was Discovered	
Manual Analysis - Pass any JS code in the input field and it will get executed.	
Vulnerable URLs	
https://www.bugbountyhunter.org/internship_labs/HTML/xss_lab/lab_1/lab_1.php	
Consequences of not Fixing the Issue	
Impersonating users and stealing credentials.	
Suggested Countermeasures	
Filter input and encode data.	
References	
https://github.com/s0md3v/AwesomeXSS	

Proof of Concept

The proof of the above vulnerability.

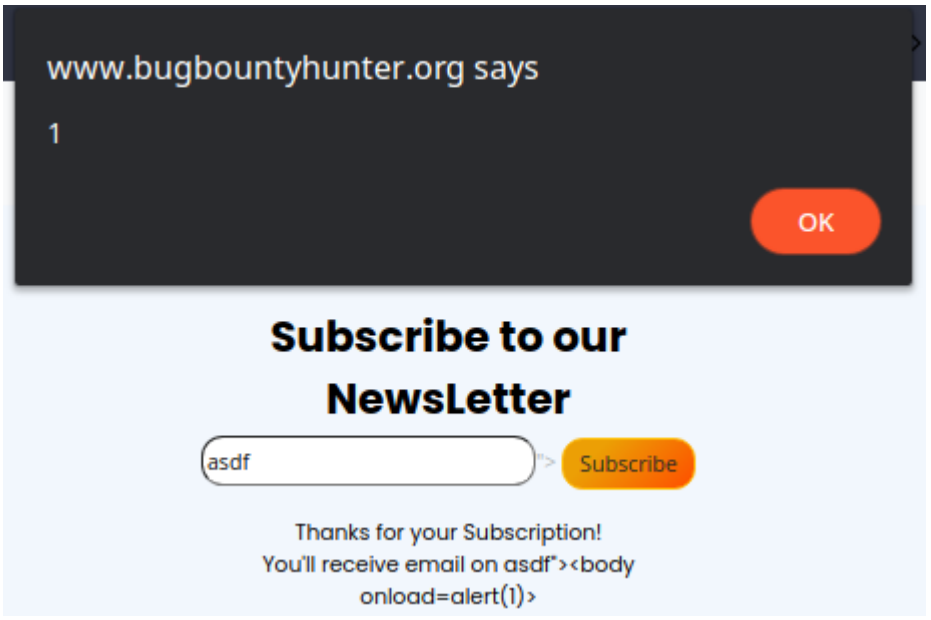


1.2. Balancing Is Important In Life!

Reference	Risk Rating
Balancing Is Important In Life!	**Low**
Tools Used	
Browser	
Vulnerability Description	
The vulnerability is Cross-Site Scripting that allows users to execute JS codes in the input fields.	
How It Was Discovered	
Manual Analysis - Pass any JS code by using a close tag or comment in the input field and it will get executed.	
Vulnerable URLs	
https://www.bugbountyhunter.org/internship_labs/HTML/xss_lab/lab_2/lab_2.php	
Consequences of not Fixing the Issue	
Impersonating users and stealing credentials.	
Suggested Countermeasures	
Filter input and encode data.	
References	
https://github.com/s0md3v/AwesomeXSS	

Proof of Concept

The proof of the above vulnerability.

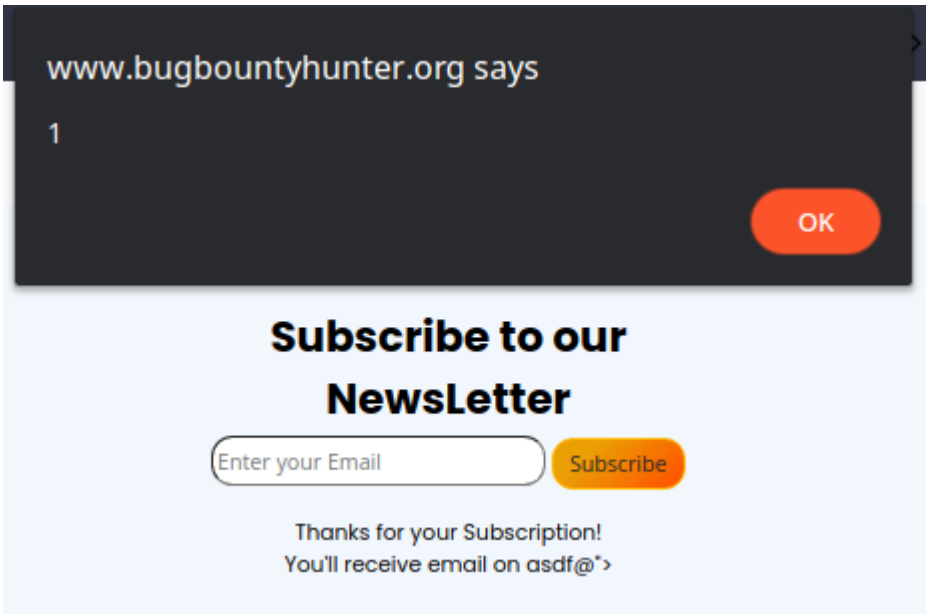


1.3. XSS Is Everywhere!

Reference	Risk Rating
XSS Is Everywhere!	**Low**
Tools Used	
Browser	
Vulnerability Description	
The vulnerability is Cross-Site Scripting that allows users to execute JS codes in the input fields.	
How It Was Discovered	
Manual Analysis - Pass any JS code by using a close tag or comment in the input field and it will get executed.	
Vulnerable URLs	
https://www.bugbountyhunter.org/internship_labs/HTML/xss_lab/lab_3/lab_3.php	
Consequences of not Fixing the Issue	
Impersonating users and stealing credentials.	
Suggested Countermeasures	
Filter input and encode data.	
References	
https://github.com/s0md3v/AwesomeXSS	

Proof of Concept

The proof of the above vulnerability.

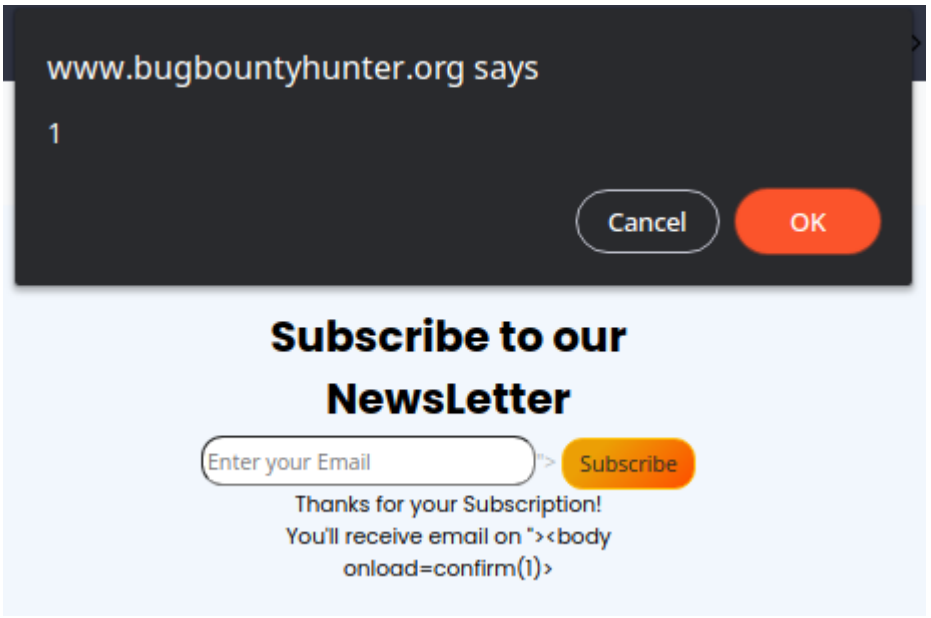


1.4. Alternative Are Must!

Reference	Risk Rating
Alternative Is Must!	**Medium**
Tools Used	
Browser	
Vulnerability Description	
The vulnerability is Cross-Site Scripting that allows users to execute JS codes in the input fields.	
How It Was Discovered	
Manual Analysis - Pass any JS code by using a close tag or comment without an alert in the input field and it will get executed.	
Vulnerable URLs	
https://www.bugbountyhunter.org/internship_labs/HTML/xss_lab/lab_4/lab_4.php	
Consequences of not Fixing the Issue	
Impersonating users and stealing credentials.	
Suggested Countermeasures	
Filter input and encode data.	
References	
https://github.com/s0md3v/AwesomeXSS	

Proof of Concept

The proof of the above vulnerability.

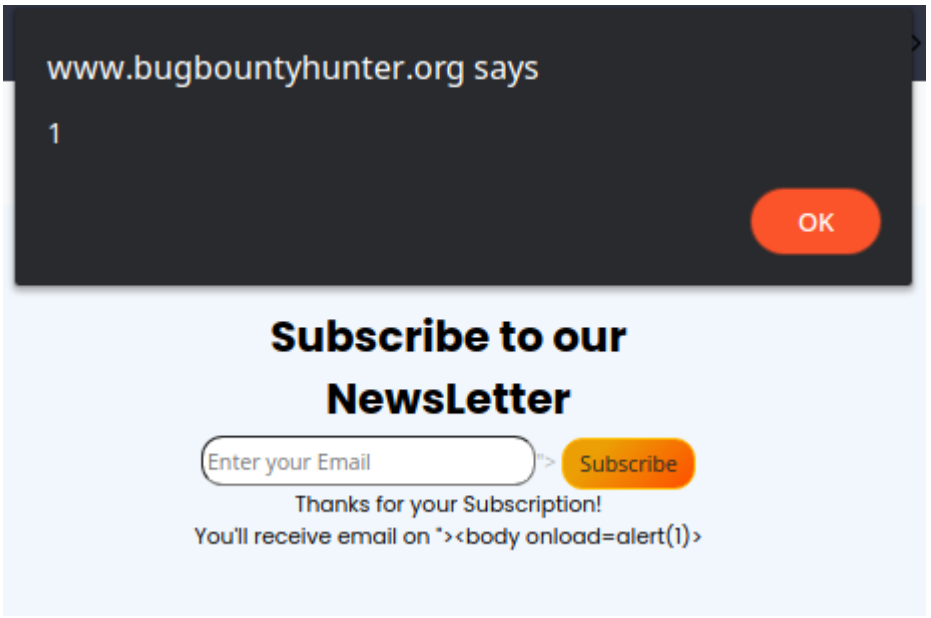


1.5. Developer Hates Scripts!

Reference	Risk Rating
Developer Hates Scripts!	**Hard**
Tools Used	
Browser	
Vulnerability Description	
The vulnerability is Cross-Site Scripting that allows users to execute JS codes in the input fields.	
How It Was Discovered	
Manual Analysis - Pass any JS code by using a close tag or comment without using script tag in the input field and it will get executed.	
Vulnerable URLs	
https://www.bugbountyhunter.org/internship_labs/HTML/xss_lab/lab_5/lab_5.php	
Consequences of not Fixing the Issue	
Impersonating users and stealing credentials.	
Suggested Countermeasures	
Filter input and encode data.	
References	
https://github.com/s0md3v/AwesomeXSS	

Proof of Concept

The proof of the above vulnerability.

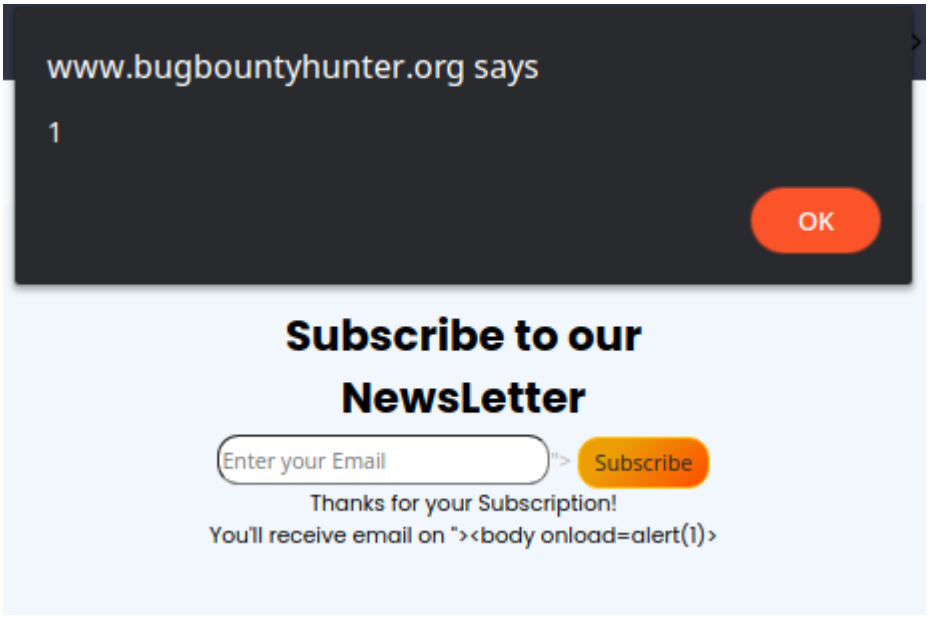


1.6. Change The Variation!

Reference	Risk Rating
Change The Variation!	**Hard**
Tools Used	
Browser	
Vulnerability Description	
The vulnerability is Cross-Site Scripting that allows users to execute JS codes in the input fields.	
How It Was Discovered	
Manual Analysis - Pass any JS code and it will get executed.	
Vulnerable URLs	
https://www.bugbountyhunter.org/internship_labs/HTML/xss_lab/lab_6/lab_6.php	
Consequences of not Fixing the Issue	
Impersonating users and stealing credentials.	
Suggested Countermeasures	
Filter input and encode data.	
References	
https://github.com/s0md3v/AwesomeXSS	

Proof of Concept

The proof of the above vulnerability.

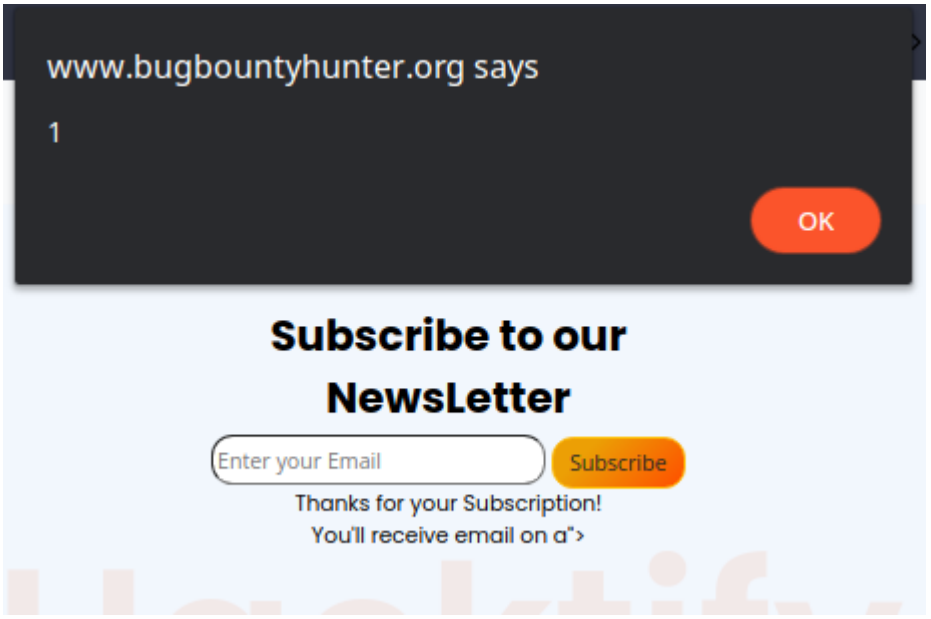


1.7. Encoding Is The Key?

Reference	Risk Rating
Encoding Is The Key?	**Medium**
Tools Used	
Browser	
Vulnerability Description	
The vulnerability is Cross-Site Scripting that allows users to execute JS codes in the input fields.	
How It Was Discovered	
Manual Analysis - Pass any URL Encoded JS code and it will get executed.	
Vulnerable URLs	
https://www.bugbountyhunter.org/internship_labs/HTML/xss_lab/lab_7/lab_7.php	
Consequences of not Fixing the Issue	
Impersonating users and stealing credentials.	
Suggested Countermeasures	
Filter input and encode data.	
References	
https://github.com/s0md3v/AwesomeXSS	

Proof of Concept

The proof of the above vulnerability.

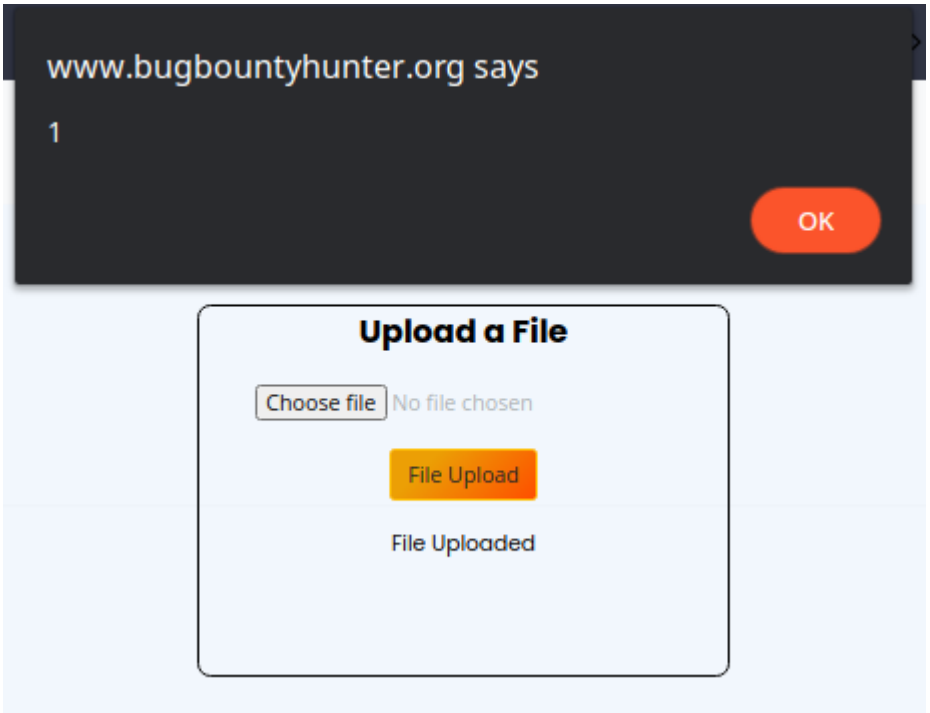


1.8. XSS With File Upload (File Name)

Reference	Risk Rating
XSS With File Upload (File Name)	**Low**
Tools Used	
Browser	
Vulnerability Description	
The vulnerability is Cross-Site Scripting that allows users to execute JS codes in the input fields.	
How It Was Discovered	
Manual Analysis - Pass any JS code as the file name, upload the file and it will get executed.	
Vulnerable URLs	
https://www.bugbountyhunter.org/internship_labs/HTML/xss_lab/lab_8/lab_8.php	
Consequences of not Fixing the Issue	
Impersonating users and stealing credentials.	
Suggested Countermeasures	
Filter input and encode data.	
References	
https://github.com/s0md3v/AwesomeXSS	

Proof of Concept

The proof of the above vulnerability.

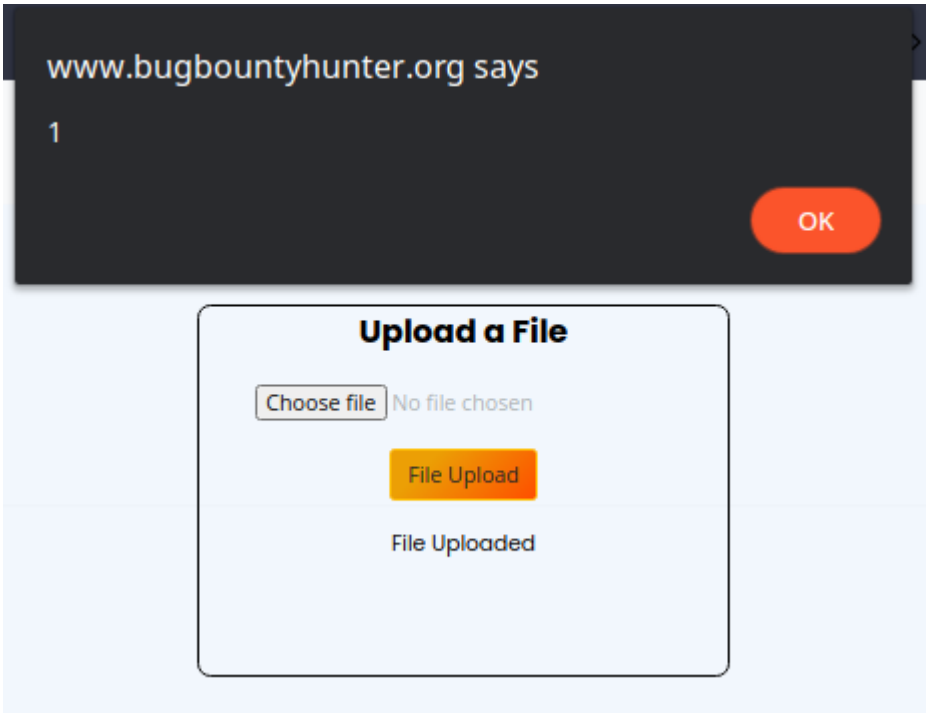


1.9. XSS With File Upload (File Content)

Reference	Risk Rating
XSS With File Upload (File Content)	**Medium**
Tools Used	
Browser	
Vulnerability Description	
The vulnerability is Cross-Site Scripting that allows users to execute JS codes in the input fields.	
How It Was Discovered	
Manual Analysis - Upload a JS file and it will get executed.	
Vulnerable URLs	
https://www.bugbountyhunter.org/internship_labs/HTML/xss_lab/lab_9/lab_9.php	
Consequences of not Fixing the Issue	
Impersonating users and stealing credentials.	
Suggested Countermeasures	
Filter input and encode data.	
References	
https://github.com/s0md3v/AwesomeXSS	

Proof of Concept

The proof of the above vulnerability.

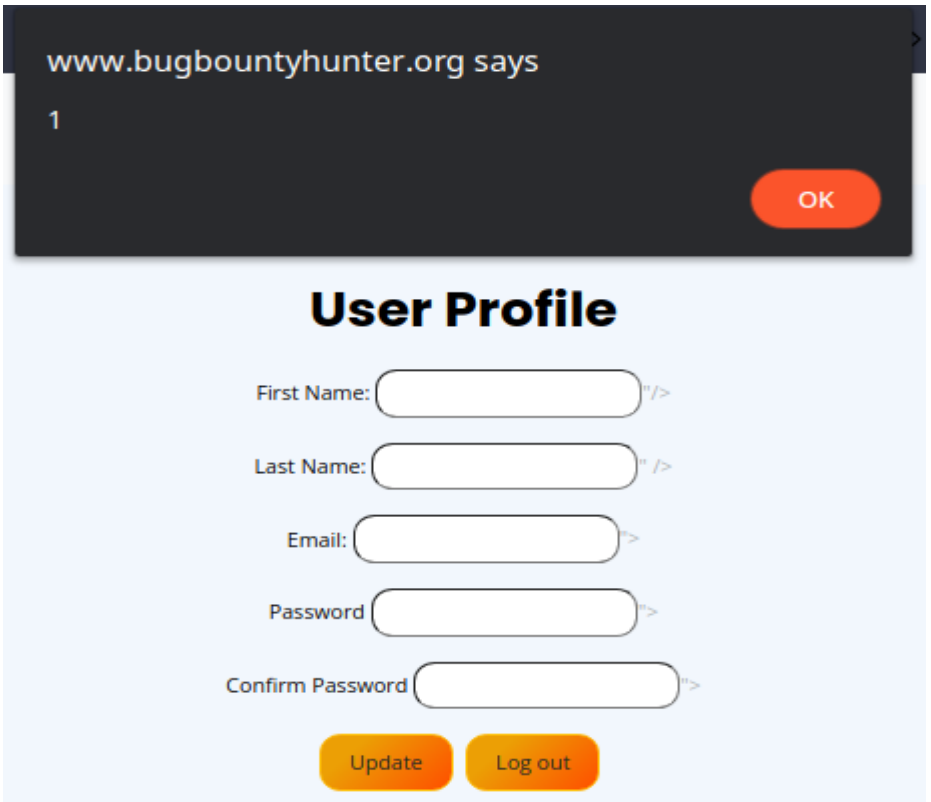


1.10. Stored Everywhere!

Reference	Risk Rating
Stored Everywhere!	**Low**
Tools Used	
Browser	
Vulnerability Description	
The vulnerability is Cross-Site Scripting that allows users to execute JS codes in the input fields.	
How It Was Discovered	
Manual Analysis - Pass any JS 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/xss_lab/lab_10/lab_10.php	
Consequences of not Fixing the Issue	
Impersonating users and stealing credentials.	
Suggested Countermeasures	
Filter input and encode data.	
References	
https://github.com/s0md3v/AwesomeXSS	

Proof of Concept

The proof of the above vulnerability.



1.11. DOM's Are Love!

Reference	Risk Rating
DOM's Are Love!	**Hard**
Tools Used	
Browser	
Vulnerability Description	
The vulnerability is Cross-Site Scripting that allows users to execute JS codes in the input fields.	
How It Was Discovered	
Manual Analysis - Pass any JS code in the input field and it will get executed.	
Vulnerable URLs	
https://www.bugbountyhunter.org/internship_labs/HTML/xss_lab/lab_11/lab_11.php	
Consequences of not Fixing the Issue	
Impersonating users and stealing credentials.	
Suggested Countermeasures	
Filter input and encode data.	
References	
https://github.com/s0md3v/AwesomeXSS	

Proof of Concept

The proof of the above vulnerability.