Penetration Testing Report

Full Name: Ilyas bajji
Program: HCPT
Date:29/02/2024

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	**Cross-Site Scripting lab, Insecure Direct Object References lab**

3. Summary

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

Total number of Sub-labs: 15 Sub-labs

High	Medium	Low
4	5	6

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**

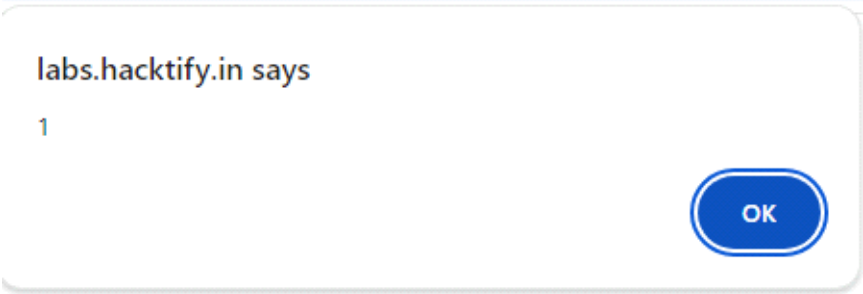
1. Cross-site scripting lab

1.1. Let's Do It!

Reference	Risk Rating
Let's do it!	Low
Tools Used	
Testing	
Vulnerability Description	
Inject javascript code to be executed whether to steal cookies or credentials of the target	
How It Was Discovered	
Manual Analysis :inject <script>alert(1)</script>	
Vulnerable URLs	
https://labs.hacktify.in/HTML/xss_lab/lab_1/lab_1.php	
Consequences of not Fixing the Issue	
tamper or steal data or doing other malicious activity	
Suggested Countermeasures	
Input Validation, Output Encoding, CSP (Content Security Policy), Browser Security Features	
References	
https://owasp.org/www-community/attacks/xss/	

Proof of Concept

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



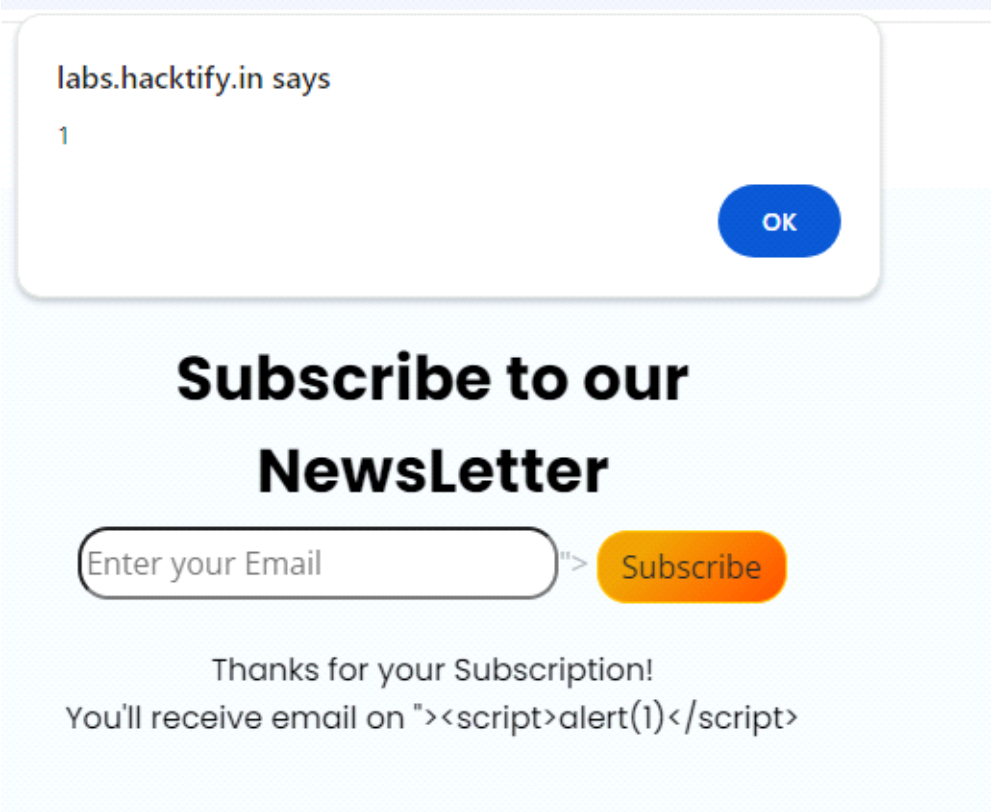
1.2. Balancing Is important in life !

Reference	Risk Rating

Balancing is important in life!	**Low**
Tools Used	
Observation and testing	
Vulnerability Description	
Inject javascript code to be executed whether to steal cookies or credentials of the target	
How It Was Discovered	
Inject : "><script>alert(1)</script>	
Vulnerable URLs	
https://labs.hacktify.in/HTML/xss_lab/lab_2/lab_2.php	
Consequences of not Fixing the Issue	
tamper or steal data or doing other malicious activity	
Suggested Countermeasures	
Input Validation, Output Encoding, CSP (Content Security Policy), Browser Security Features	
References	
https://owasp.org/www-community/attacks/xss/	

Proof of Concept

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

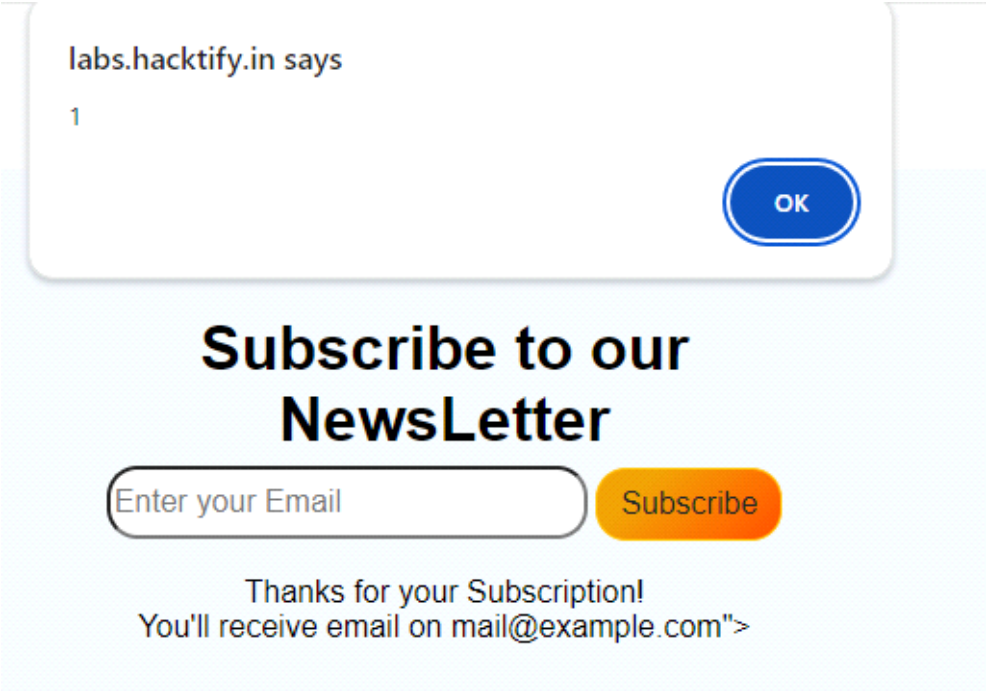


1.3. XSS is everywhere

Reference	Risk Rating
XSS is everywhere	**Low**
Tools Used	
Observation and testing	
Vulnerability Description	
Inject javascript code to be executed whether to steal cookies or credentials of the target	
How It Was Discovered	
Inject : mail@example.com"><script>alert(1)</script>	
Vulnerable URLs	
https://labs.hacktify.in/HTML/xss_lab/lab_2/lab_2.php	
Consequences of not Fixing the Issue	
tamper or steal data or doing other malicious activity	
Suggested Countermeasures	
Input Validation, Output Encoding, CSP (Content Security Policy), Browser Security Features	
References	
https://owasp.org/www-community/attacks/xss/	

Proof of Concept

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

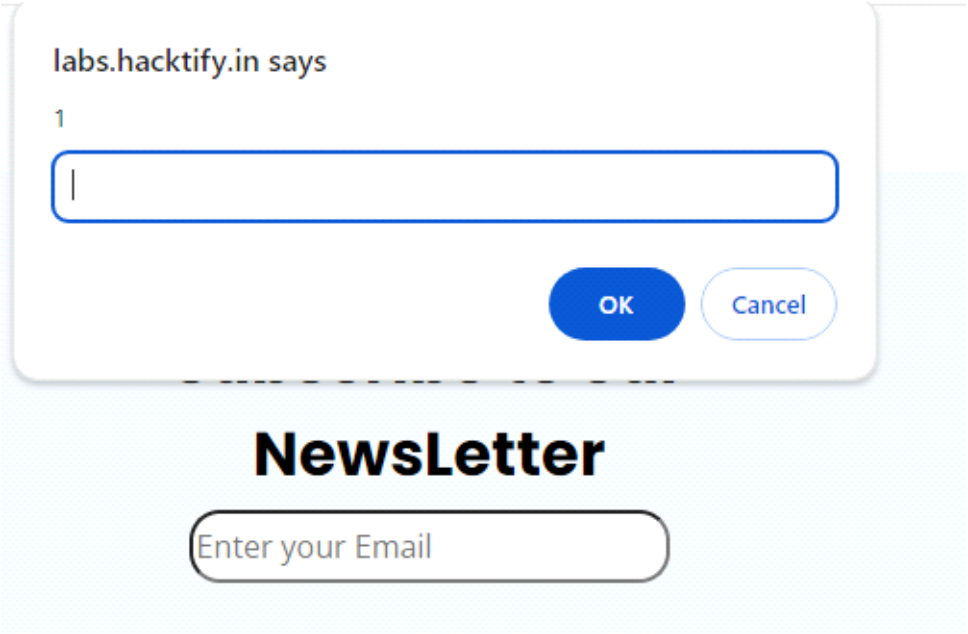


1.4. Alternatives are must

Reference	Risk Rating
Alternatives are must	**Medium**
Tools Used	
Observation and testing	
Vulnerability Description	
Inject javascript code to be executed whether to steal cookies or credentials of the target	
How It Was Discovered	
Inject : "><script>prompt("1")</script>	
Vulnerable URLs	
https://labs.hacktify.in/HTML/xss_lab/lab_4/lab_4.php	
Consequences of not Fixing the Issue	
tamper or steal data or doing other malicious activity	
Suggested Countermeasures	
Input Validation, Output Encoding, CSP (Content Security Policy), Browser Security Features	
References	
https://owasp.org/www-community/attacks/xss/	

Proof of Concept

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

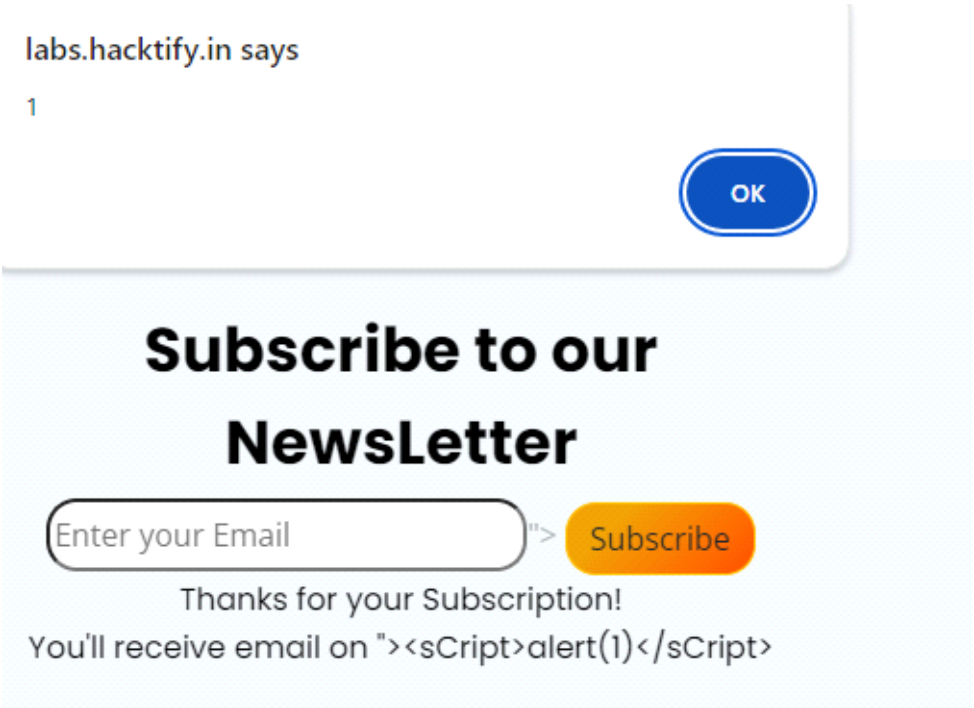


1.5.Developer Hates Scripts!

Reference	Risk Rating
Developer hates scripts!	**Medium**
Tools Used	
Observation and testing	
Vulnerability Description	
Inject javascript code to be executed whether to steal cookies or credentials of the target	
How It Was Discovered	
Inject : "><sCript>alert(1)</sCript>	
Vulnerable URLs	
https://labs.hacktify.in/HTML/xss_lab/lab_5/lab_5.php	
Consequences of not Fixing the Issue	
tamper or steal data or doing other malicious activity	
Suggested Countermeasures	
Input Validation, Output Encoding, CSP (Content Security Policy), Browser Security Features	
References	
https://owasp.org/www-community/attacks/xss/	

Proof of Concept

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

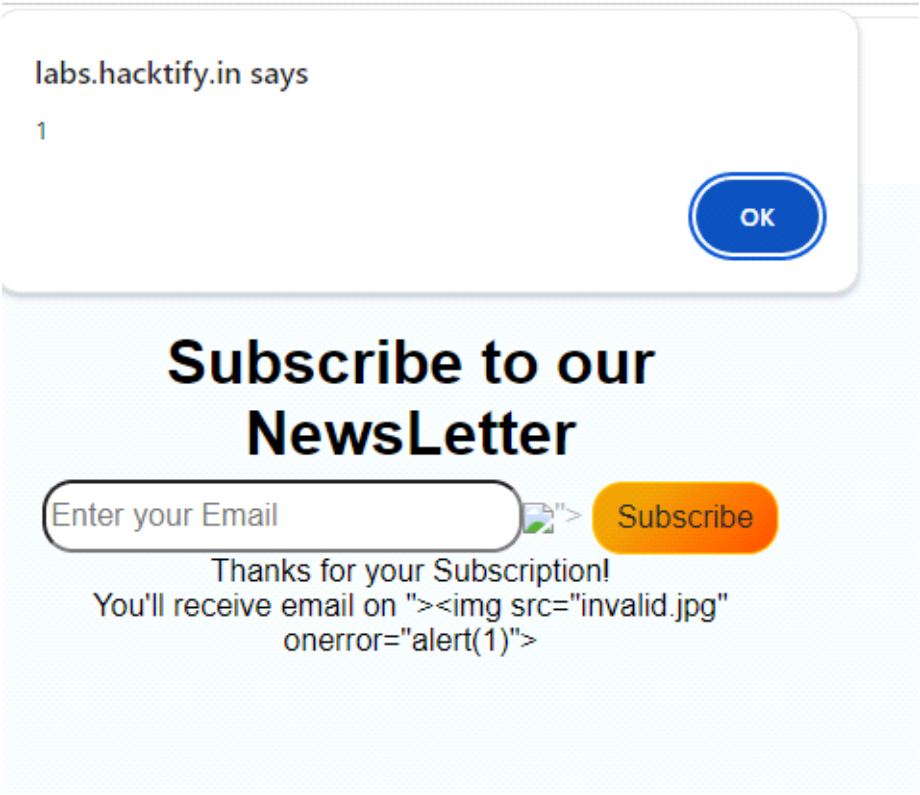


1.6. Change the variation!

Reference	Risk Rating
Change the variation	**Medium**
Tools Used	
Observation and testing	
Vulnerability Description	
Inject javascript code to be executed whether to steal cookies or credentials of the target	
How It Was Discovered	
Inject : "><img src="invalid.jpg" onerror="alert(1)">	
Vulnerable URLs	
https://labs.hacktify.in/HTML/xss_lab/lab_6/lab_6.php	
Consequences of not Fixing the Issue	
tamper or steal data or doing other malicious activity	
Suggested Countermeasures	
Input Validation, Output Encoding, CSP (Content Security Policy), Browser Security Features	
References	
https://owasp.org/www-community/attacks/xss/	

Proof of Concept

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

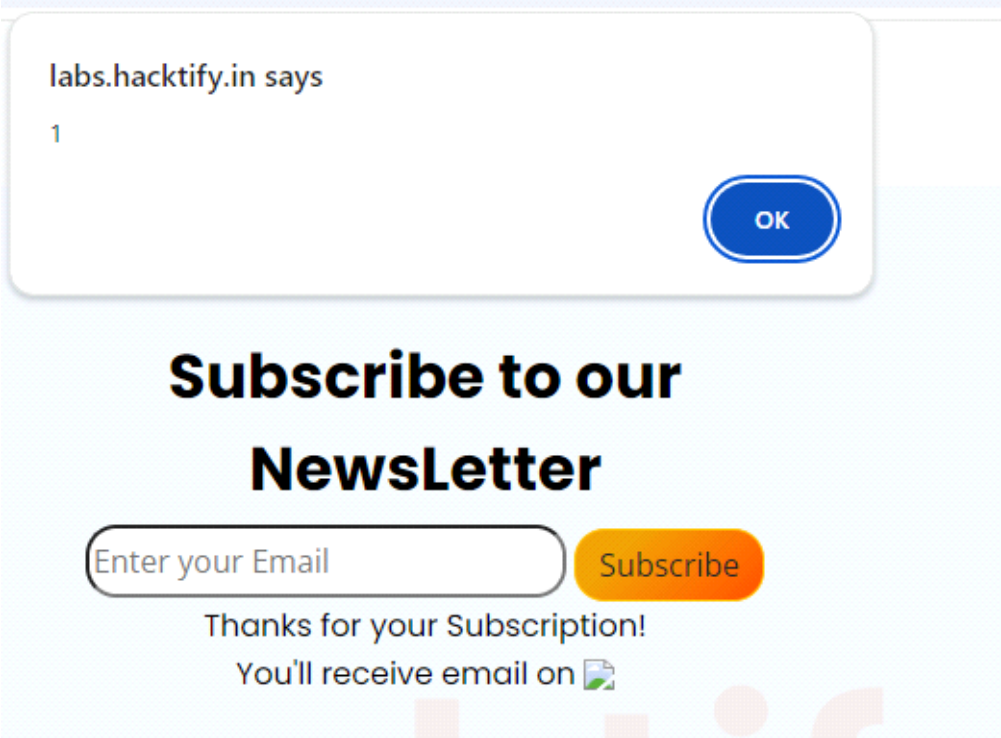


1.7. Encoding is the key

Reference	Risk Rating
Balancing is important in life!	**Low**
Tools Used	
Observation and testing	
Vulnerability Description	
Inject javascript code to be executed whether to steal cookies or credentials of the target	
How It Was Discovered	
Inject : %3Cimg src%3Dx onerror%3Dalert%28"1"%29%3E	
Vulnerable URLs	
https://labs.hacktify.in/HTML/xss_lab/lab_7/lab_7.php	
Consequences of not Fixing the Issue	
tamper or steal data or doing other malicious activity	
Suggested Countermeasures	
Input Validation, Output Encoding, CSP (Content Security Policy), Browser Security Features	
References	
https://owasp.org/www-community/attacks/xss/	

Proof of Concept

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

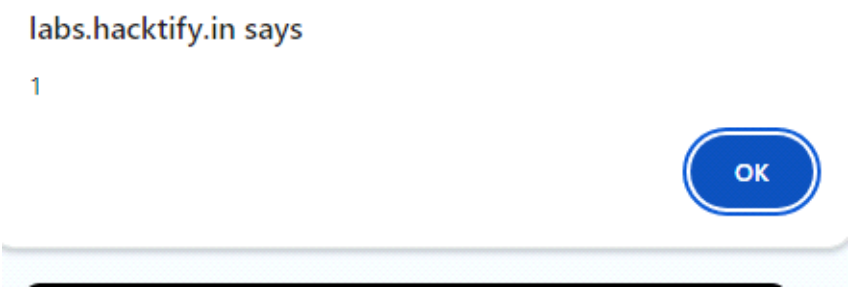


1.8. XSS with File upload (file name)

Reference	Risk Rating
XSS with File upload (file name)	**Low**
Tools Used	
Observation and testing	
Vulnerability Description	
Inject javascript code to be executed whether to steal cookies or credentials of the target	
How It Was Discovered	
Inject : <img src=x onerror=alert(1)>	
Vulnerable URLs	
https://labs.hacktify.in/HTML/xss_lab/lab_8/lab_8.php	
Consequences of not Fixing the Issue	
tamper or steal data or doing other malicious activity	
Suggested Countermeasures	
Input Validation, Output Encoding, CSP (Content Security Policy), Browser Security Features	
References	
https://owasp.org/www-community/attacks/xss/	

Proof of Concept

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

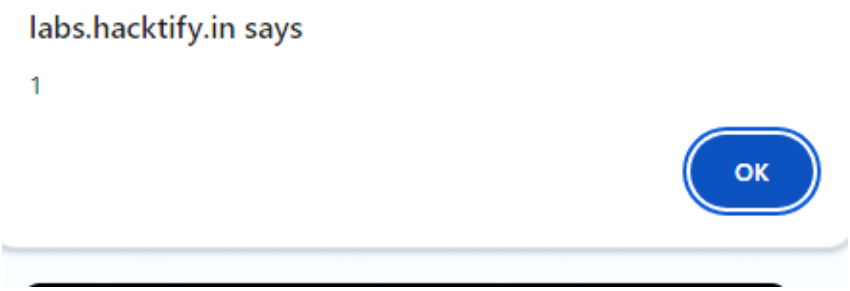


1.9. XSS with file upload (file content)

Reference	Risk Rating
XSS with file upload (file content)	**medium**
Tools Used	
Observation and testing	
Vulnerability Description	
Inject javascript code to be executed whether to steal cookies or credentials of the target	
How It Was Discovered	
Inject : <script>alert(1)</script>	
Vulnerable URLs	
https://labs.hacktify.in/HTML/xss_lab/lab_9/lab_9.php	
Consequences of not Fixing the Issue	
tamper or steal data or doing other malicious activity	
Suggested Countermeasures	
Input Validation, Output Encoding, CSP (Content Security Policy), Browser Security Features	
References	
https://owasp.org/www-community/attacks/xss/	

Proof of Concept

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

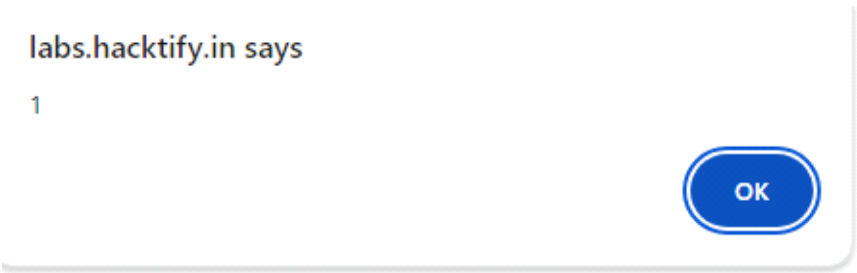


1.10. Stored Everywhere

Reference	Risk Rating
Stored everywhere	**Low**
Tools Used	
Observation and testing	
Vulnerability Description	
Inject javascript code to be executed whether to steal cookies or credentials of the target	
How It Was Discovered	
Inject : "><script>alert(1)</script>	
Vulnerable URLs	
https://labs.hacktify.in/HTML/xss_lab/lab_10/lab_10.php	
Consequences of not Fixing the Issue	
tamper or steal data or doing other malicious activity	
Suggested Countermeasures	
Input Validation, Output Encoding, CSP (Content Security Policy), Browser Security Features	
References	
https://owasp.org/www-community/attacks/xss/	

Proof of Concept

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

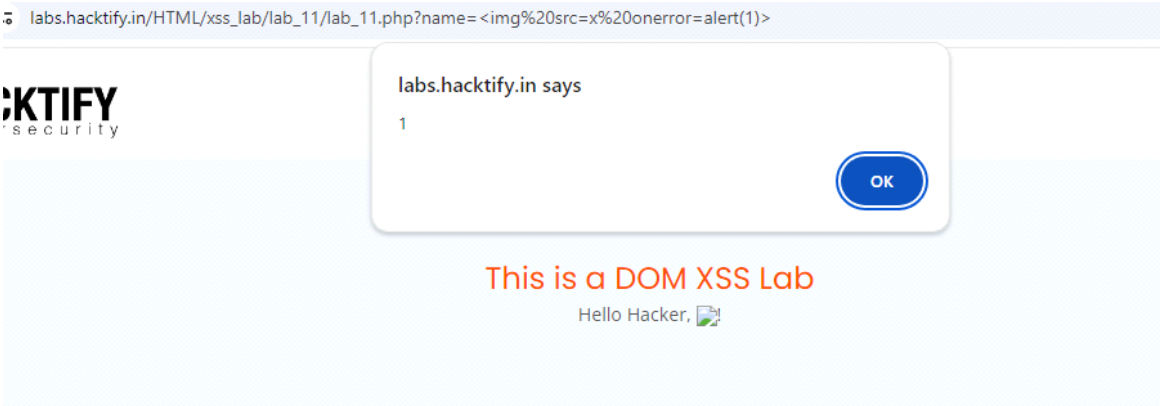


1.11. DOM's are love !

Reference	Risk Rating
DOM's are love!	**high**
Tools Used	
Observation and testing	
Vulnerability Description	
Inject javascript code to be executed whether to steal cookies or credentials of the target	
How It Was Discovered	
Inject : ?name=<img%20src=x%20onerror=alert(1)>	
Vulnerable URLs	
https://labs.hacktify.in/HTML/xss_lab/lab_11/lab_11.php	
Consequences of not Fixing the Issue	
tamper or steal data or doing other malicious activity	
Suggested Countermeasures	
Input Validation, Output Encoding, CSP (Content Security Policy), Browser Security Features	
References	
https://owasp.org/www-community/attacks/xss/	

Proof of Concept

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



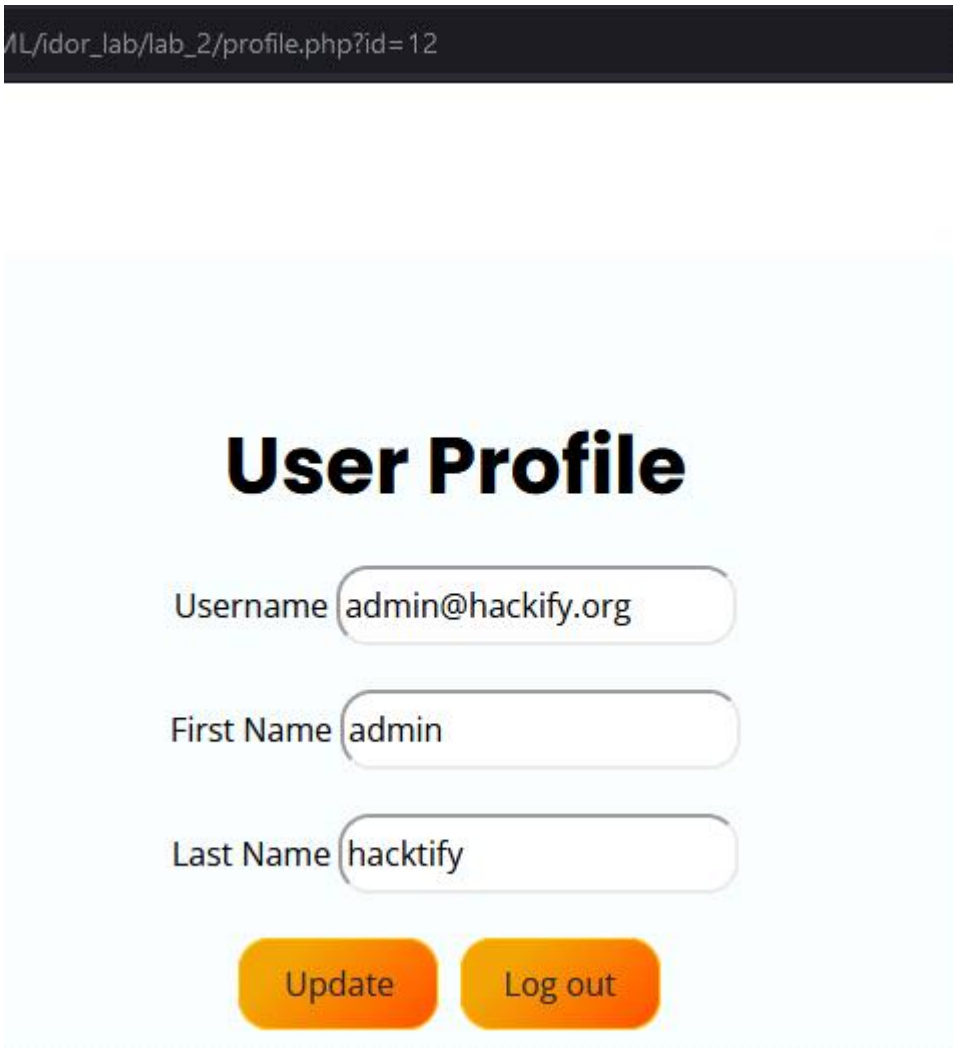
2.IDOR

2.1. Give me my amount!!

Reference	Risk Rating
Give me my amount	**Low**
Tools Used	

id parameter in url
Vulnerability Description
we can login to any account
How It Was Discovered
Manual Analysis:id=12 any onther id
Vulnerable URLs
https://labs.hacktify.in/HTML/idor_lab/lab_1/profile.php?id=12
Consequences of not Fixing the Issue
we can access someone else's account
Suggested Countermeasures
remove the id paramter from the url
References
https://www.varonis.com/blog/what-is-idor-insecure-direct-object-reference

Proof of Concept

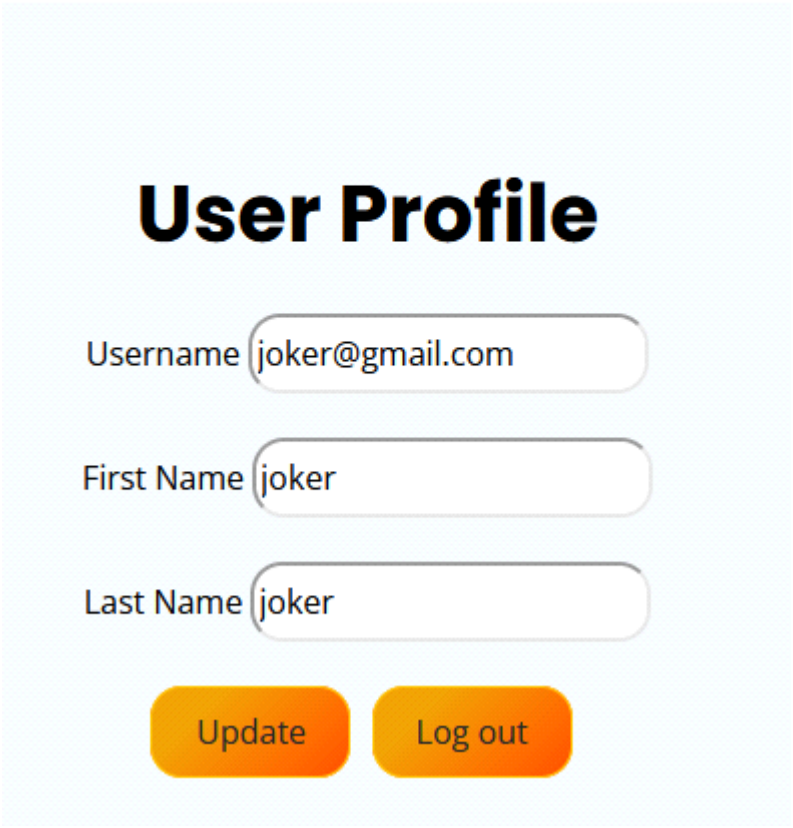


2.2. Stop polluting my params!

Reference	Risk Rating
Stop polluting my params!	**Medium**
Tools Used	
id parameter in url	
Vulnerability Description	
we can login to any account	
How It Was Discovered	
Manual Analysis:id=26 any onther id	
Vulnerable URLs	
https://labs.hacktify.in/HTML/idor_lab/lab_1/profile.php?id=26	
Consequences of not Fixing the Issue	
we can access someone else's account	

Suggested Countermeasures
remove the id paramter from the url
References
https://www.varonis.com/blog/what-is-idor-insecure-direct-object-reference

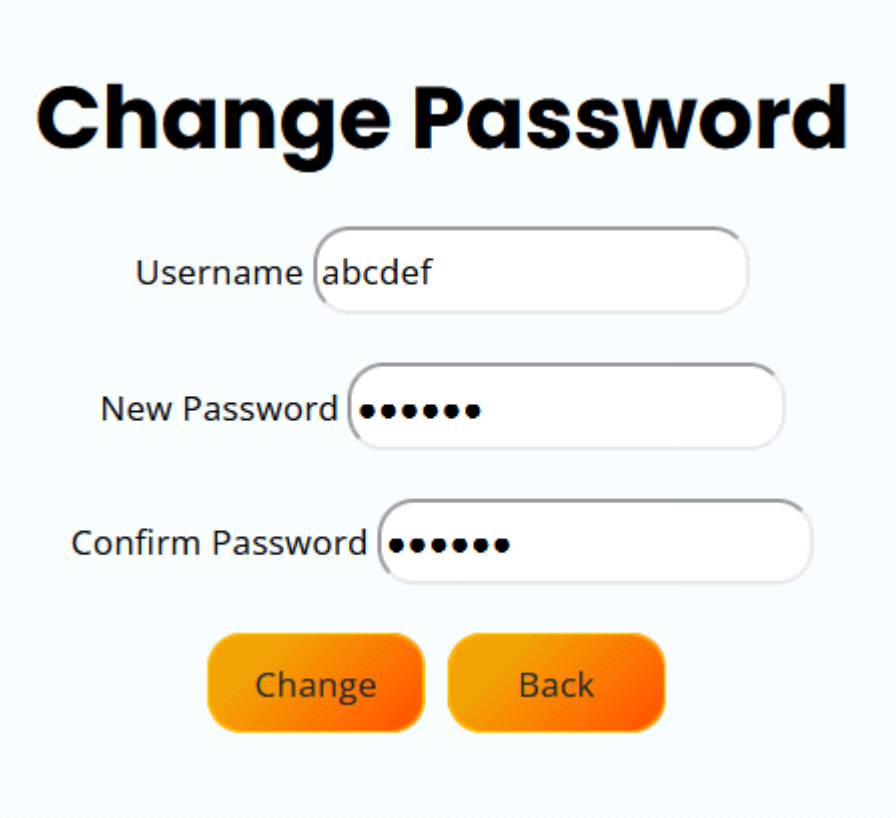
Proof of Concept



2.3. Someone changed my Password

Reference	Risk Rating
Someone changed my Password	**High**
Tools Used	
I register with the victim's email address	
Vulnerability Description	
We can change the password easily	
How It Was Discovered	
Manual Analysis: we can change the password, then we can use the account	
Vulnerable URLs	
https://labs.hacktify.in/HTML/idor_lab/lab_3/changepassword.php?username=abcdef	
Consequences of not Fixing the Issue	
we can use any account in this bank	
Suggested Countermeasures	
send an verification email to the user in his mail or phone	
References	
https://www.invicti.com/learn/insecure-direct-object-references-idor/	

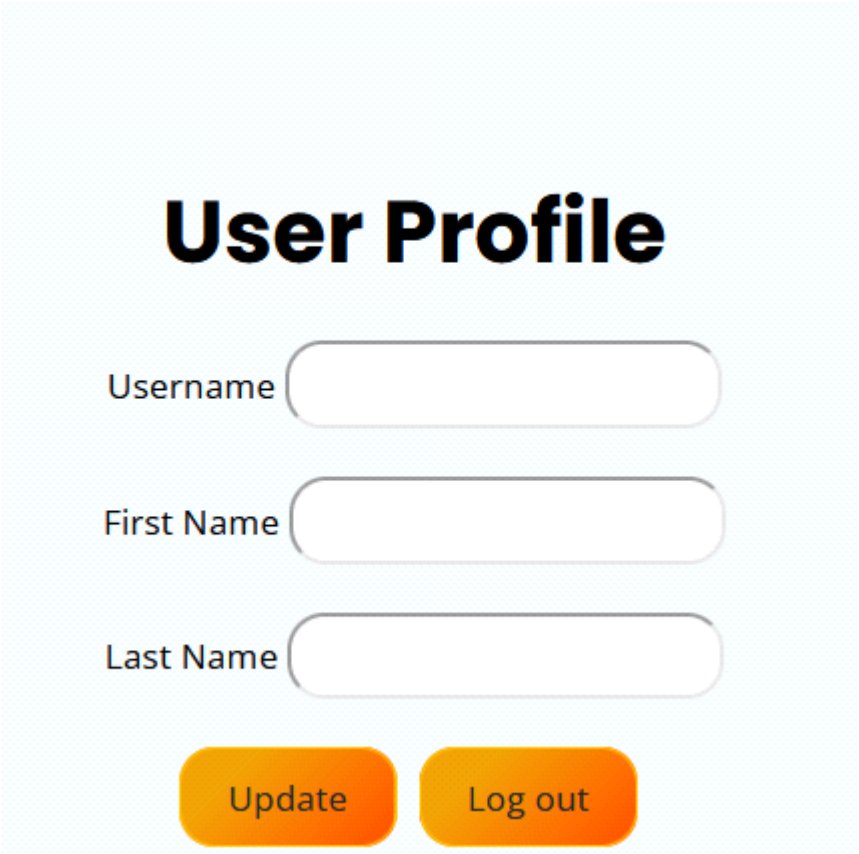
Proof of Concept



2.4. Change your methods!

Reference	Risk Rating
Change your methods	**Medium**
Tools Used	
I register with the victim's email address	
Vulnerability Description	
We can do register an any account	
How It Was Discovered	
Manual Analysis: we can registeran any acount, then we can use the account	
Vulnerable URLs	
https://labs.hacktify.in/HTML/idor_lab/lab_4/profile.php?id=509	
Consequences of not Fixing the Issue	
we can register an any account in this bank then we will have the accessebelity to the account	
Suggested Countermeasures	
send an verification email to the user in his mail or phone	
References	
https://www.invicti.com/learn/insecure-direct-object-references-idor/	

Proof of Concept


User Profile
Username
First Name
Last Name
Update
Log out