Penetration Testing Report

Full Name: Muhammad Bokhtiar Uddin
Program: HCS - Penetration Testing Internship Week-2
Date: 28/02/2024

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 {#} 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, Insecure Direct Object References

3. Summary

Outlined is a Black Box Application Security assessment for the **Week {#} Labs**.

Total number of Sub-labs: {count} Sub-labs

High	Medium	Low
{count}	{count}	{count}

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**

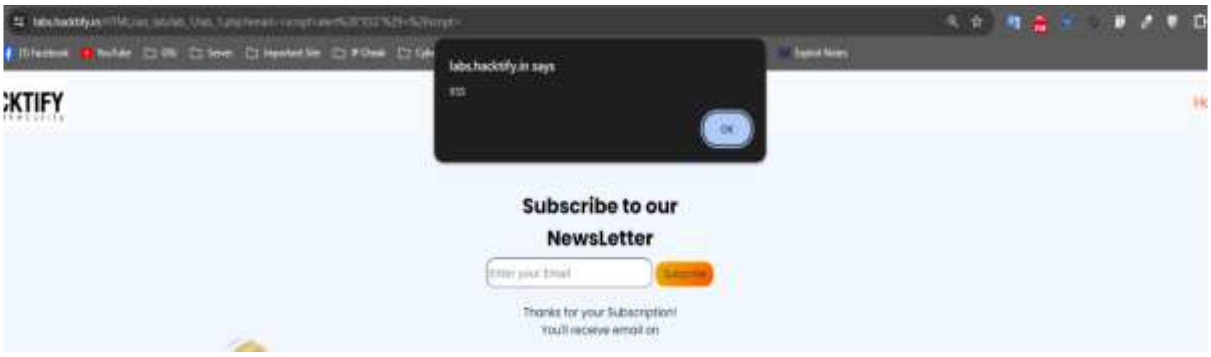
1. Cross Site Scripting!

1.1. Let's Do IT!

Reference	Risk Rating
Let's Do It!	**Low / Medium / High**
Tools Used	
Burp Suit	
Vulnerability Description	
XSS is a web app flaw where attackers inject harmful scripts into pages viewed by users. These scripts steal data, hijack sessions, or change page content. Exploited in input fields, like search boxes, they let attackers run code in victims' browsers.	
How It Was Discovered	
Manual Analysis	
Vulnerable URLs	
https://labs.hacktify.in/HTML/xss_lab/lab_1/lab_1.php	
Consequences of not Fixing the Issue	
Data Theft: Attackers can steal sensitive user information like passwords and credit card details. **Session Hijacking:** They can take control of user sessions, allowing unauthorized actions on behalf of users. **Phishing Attacks:** Creation of convincing phishing links to trick users into revealing credentials or downloading malware. **Malware Distribution:** Injection of code to automatically download and execute malware on users' devices. **Reputation Damage:** Loss of user trust, site traffic, and potential customers due to security risks.	
Suggested Countermeasures	
Input Validation: Validate and sanitize user input to ensure it meets expected formats and limits. **Output Encoding:** Encode user-generated content before displaying it to prevent script execution. **HTTP Only Cookies:** Use HTTP Only flag for cookies to prevent client-side script access. **Content Security Policy (CSP):** Implement CSP to restrict sources of executable scripts.	
References	
https://portswigger.net/web-security/cross-site-scripting	

Proof of Concept

Payload: <script>alert("XSS")</script>



labs.hacktify.in says

XSS

Subscribe to our NewsLetter

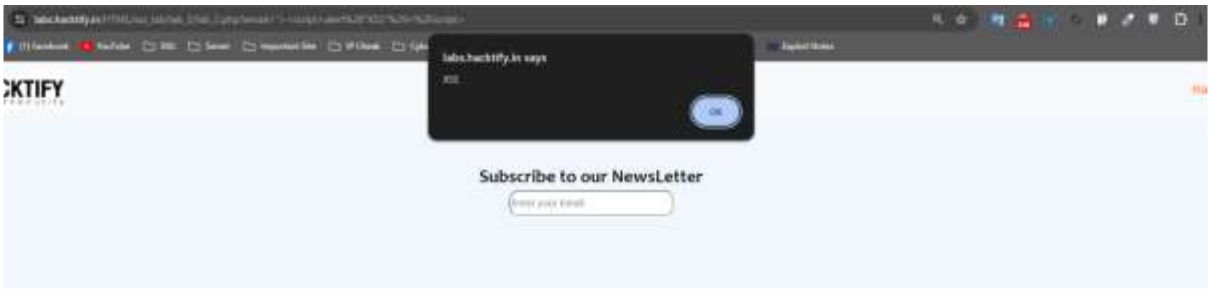
Thanks for your Subscription!
You'll receive email on

1.2. Balancing Is Important In Life!

Reference	**Risk Rating**
Balancing Is Important in Life!	**Low / Medium / High**
Tools Used	
Burp Suite	
Vulnerability Description	
XSS is a web app flaw where attackers inject harmful scripts into pages viewed by users. These scripts steal data, hijack sessions, or change page content. Exploited in input fields, like search boxes, they let attackers run code in victims' browsers.	
How It Was Discovered	
Manual Analysis	
Vulnerable URLs	
https://labs.hacktify.in/HTML/xss_lab/lab_2/lab_2.php	
Consequences of not Fixing the Issue	
Data Theft: Attackers can steal sensitive user information like passwords and credit card details. **Session Hijacking:** They can take control of user sessions, allowing unauthorized actions on behalf of users. **Phishing Attacks:** Creation of convincing phishing links to trick users into revealing credentials or downloading malware. **Malware Distribution:** Injection of code to automatically download and execute malware on users' devices. **Reputation Damage:** Loss of user trust, site traffic, and potential customers due to security risks.	
Suggested Countermeasures	
Input Validation: Validate and sanitize user input to ensure it meets expected formats and limits. **Output Encoding:** Encode user-generated content before displaying it to prevent script execution. **HTTP Only Cookies:** Use HTTP Only flag for cookies to prevent client-side script access. **Content Security Policy (CSP):** Implement CSP to restrict sources of executable scripts.	
References	
https://portswigger.net/web-security/cross-site-scripting	

Proof of Concept

Payload: “><script>alert("XSS")</script>


labs.hacktify.in says

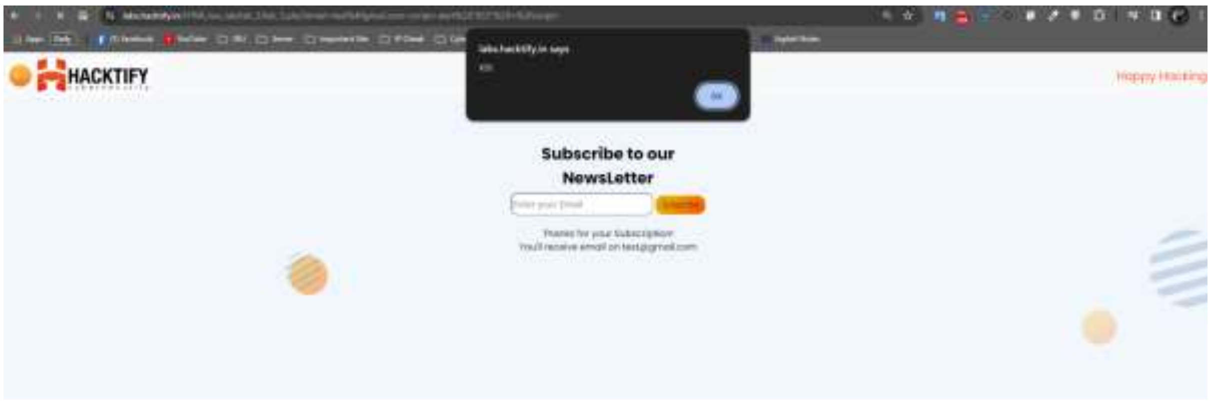
XSS

1.3. XSS Is Everywhere!

Reference	Risk Rating
XSS Is Everywhere!	**Low / Medium / High**
Tools Used	
Burp Suite	
Vulnerability Description	
XSS is a web app flaw where attackers inject harmful scripts into pages viewed by users. These scripts steal data, hijack sessions, or change page content. Exploited in input fields, like search boxes, they let attackers run code in victims' browsers.	
How It Was Discovered	
Manual Analysis	
Vulnerable URLs	
https://labs.hacktify.in/HTML/xss_lab/lab_3/lab_3.php	
Consequences of not Fixing the Issue	
Data Theft: Attackers can steal sensitive user information like passwords and credit card details. **Session Hijacking:** They can take control of user sessions, allowing unauthorized actions on behalf of users. **Phishing Attacks:** Creation of convincing phishing links to trick users into revealing credentials or downloading malware. **Malware Distribution:** Injection of code to automatically download and execute malware on users' devices. **Reputation Damage:** Loss of user trust, site traffic, and potential customers due to security risks.	
Suggested Countermeasures	
Input Validation: Validate and sanitize user input to ensure it meets expected formats and limits. **Output Encoding:** Encode user-generated content before displaying it to prevent script execution. **HTTP Only Cookies:** Use HTTP Only flag for cookies to prevent client-side script access. **Content Security Policy (CSP):** Implement CSP to restrict sources of executable scripts.	
References	
https://portswigger.net/web-security/cross-site-scripting	

Proof of Concept

Payload: 0xbt@gmail.com<script>alert(”XSS”)</script>


labs.hacktify.in says

XSS

Subscribe to our NewsLetter

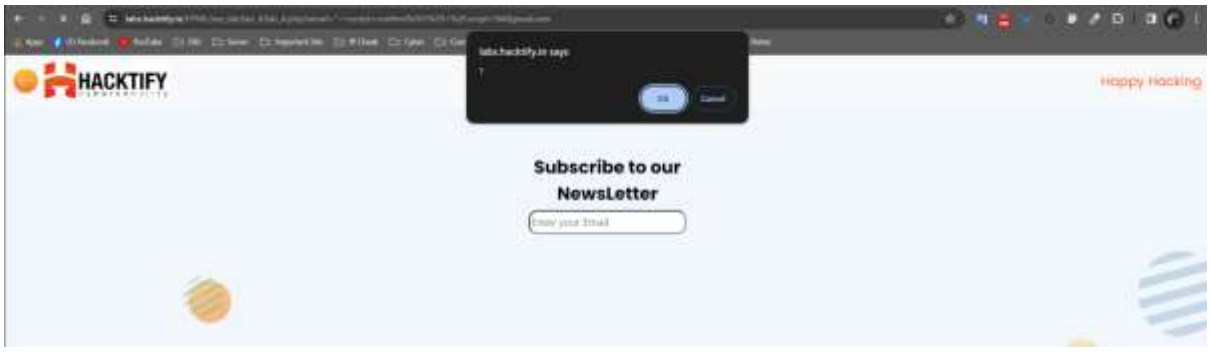
Thanks for your Subscription
You'll receive email on test@gmail.com

1.4. Alternatives Are Must!

Reference	Risk Rating
Alternatives Are Must!	Low / **Medium** / High
Tools Used	
Burp Suite	
Vulnerability Description	
XSS is a web app flaw where attackers inject harmful scripts into pages viewed by users. These scripts steal data, hijack sessions, or change page content. Exploited in input fields, like search boxes, they let attackers run code in victims' browsers.	
How It Was Discovered	
Manual Analysis	
Vulnerable URLs	
https://labs.hacktify.in/HTML/xss_lab/lab_4/lab_4.php	
Consequences of not Fixing the Issue	
Data Theft: Attackers can steal sensitive user information like passwords and credit card details. **Session Hijacking:** They can take control of user sessions, allowing unauthorized actions on behalf of users. **Phishing Attacks:** Creation of convincing phishing links to trick users into revealing credentials or downloading malware. **Malware Distribution:** Injection of code to automatically download and execute malware on users' devices. **Reputation Damage:** Loss of user trust, site traffic, and potential customers due to security risks.	
Suggested Countermeasures	
Input Validation: Validate and sanitize user input to ensure it meets expected formats and limits. **Output Encoding:** Encode user-generated content before displaying it to prevent script execution. **HTTP Only Cookies:** Use HTTP Only flag for cookies to prevent client-side script access. **Content Security Policy (CSP):** Implement CSP to restrict sources of executable scripts.	
References	
https://portswigger.net/web-security/cross-site-scripting	

Proof of Concept

Payload: "><script>confirm(1)</script>0xbt@gmail.com

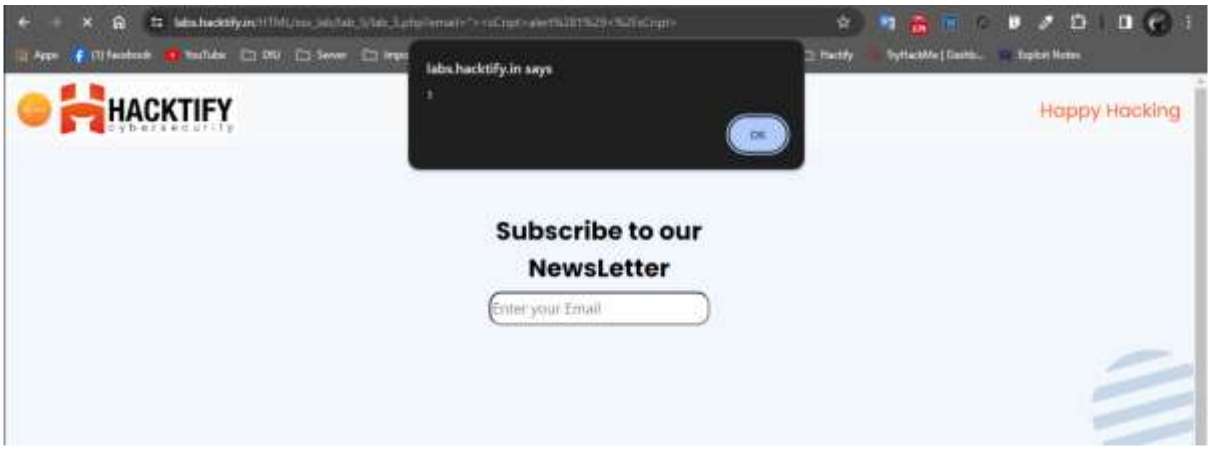


1.5. Developer Hates Scripts!

Reference	**Risk Rating**
Developer Hates Script!	**Low / Medium / High**
Tools Used	
Burp Suite	
Vulnerability Description	
XSS is a web app flaw where attackers inject harmful scripts into pages viewed by users. These scripts steal data, hijack sessions, or change page content. Exploited in input fields, like search boxes, they let attackers run code in victims' browsers.	
How It Was Discovered	
Manual Analysis	
Vulnerable URLs	
https://labs.hacktify.in/HTML/xss_lab/lab_5/lab_5.php	
Consequences of not Fixing the Issue	
Data Theft: Attackers can steal sensitive user information like passwords and credit card details. **Session Hijacking:** They can take control of user sessions, allowing unauthorized actions on behalf of users. **Phishing Attacks:** Creation of convincing phishing links to trick users into revealing credentials or downloading malware. **Malware Distribution:** Injection of code to automatically download and execute malware on users' devices. **Reputation Damage:** Loss of user trust, site traffic, and potential customers due to security risks.	
Suggested Countermeasures	
Input Validation: Validate and sanitize user input to ensure it meets expected formats and limits. **Output Encoding:** Encode user-generated content before displaying it to prevent script execution. **HTTP Only Cookies:** Use HTTP Only flag for cookies to prevent client-side script access. **Content Security Policy (CSP):** Implement CSP to restrict sources of executable scripts.	
References	
https://portswigger.net/web-security/cross-site-scripting	

Proof of Concept

Payload: "><sCript>alert(1)</sCript>


labs.hacktify.in says

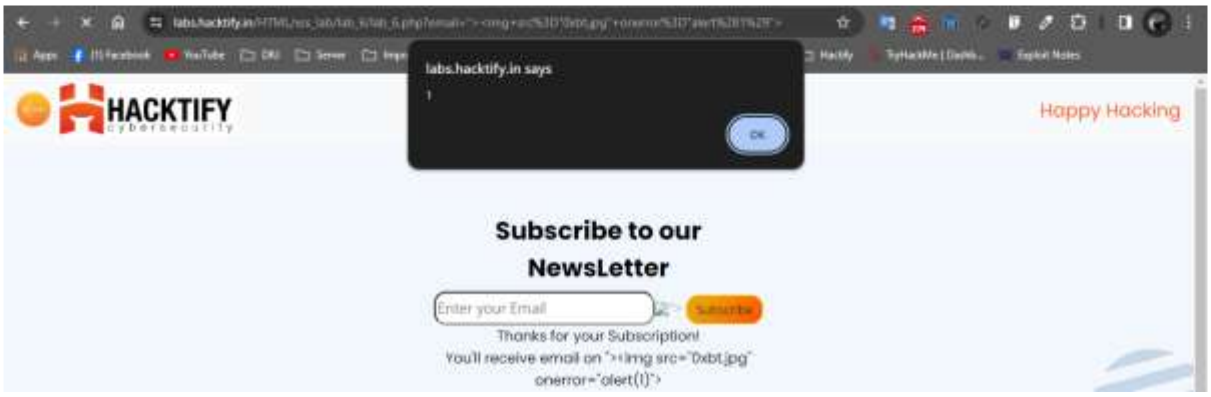
1

1.6. Change The Variation!

Reference	Risk Rating
Change The Variation!	**Low / Medium / High**
Tools Used	
Burp Suite	
Vulnerability Description	
XSS is a web app flaw where attackers inject harmful scripts into pages viewed by users. These scripts steal data, hijack sessions, or change page content. Exploited in input fields, like search boxes, they let attackers run code in victims' browsers.	
How It Was Discovered	
Manual Analysis	
Vulnerable URLs	
https://labs.hacktify.in/HTML/xss_lab/lab_6/lab_6.php	
Consequences of not Fixing the Issue	
Data Theft: Attackers can steal sensitive user information like passwords and credit card details. **Session Hijacking:** They can take control of user sessions, allowing unauthorized actions on behalf of users. **Phishing Attacks:** Creation of convincing phishing links to trick users into revealing credentials or downloading malware. **Malware Distribution:** Injection of code to automatically download and execute malware on users' devices. **Reputation Damage:** Loss of user trust, site traffic, and potential customers due to security risks.	
Suggested Countermeasures	
Input Validation: Validate and sanitize user input to ensure it meets expected formats and limits. **Output Encoding:** Encode user-generated content before displaying it to prevent script execution. **HTTP Only Cookies:** Use HTTP Only flag for cookies to prevent client-side script access. **Content Security Policy (CSP):** Implement CSP to restrict sources of executable scripts.	
References	
https://portswigger.net/web-security/cross-site-scripting	

Proof of Concept

Payload: "><img src="0xbt.jpg" onerror="alert(1)">



labs.hacktify.in says

1

Subscribe to our NewsLetter

Thanks for your Subscription!
You'll receive email on "><img src="0xbt.jpg" onerror="alert(1)">

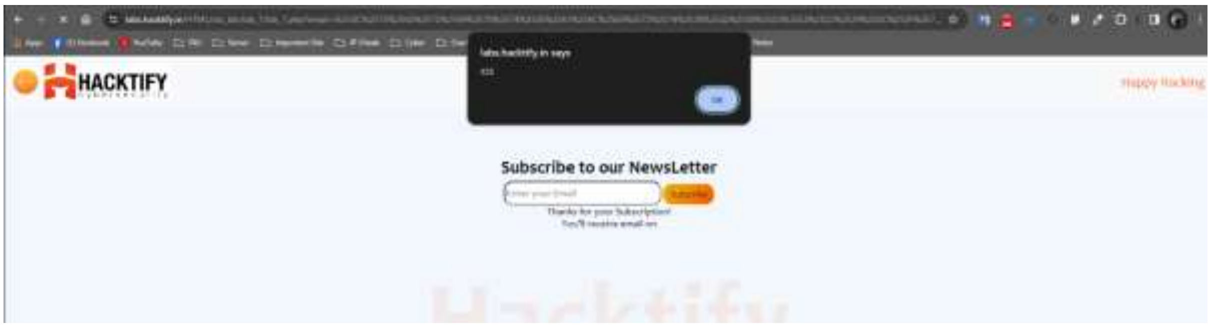
1.7. Encoding Is The Key?

Reference	**Risk Rating**
Encoding Is The Key?	**Low / Medium / High**
Tools Used	
Burp Suite	
Vulnerability Description	
XSS is a web app flaw where attackers inject harmful scripts into pages viewed by users. These scripts steal data, hijack sessions, or change page content. Exploited in input fields, like search boxes, they let attackers run code in victims' browsers.	
How It Was Discovered	
Manual Analysis	
Vulnerable URLs	
https://labs.hacktify.in/HTML/xss_lab/lab_7/lab_7.php	
Consequences of not Fixing the Issue	
Data Theft: Attackers can steal sensitive user information like passwords and credit card details. **Session Hijacking:** They can take control of user sessions, allowing unauthorized actions on behalf of users. **Phishing Attacks:** Creation of convincing phishing links to trick users into revealing credentials or downloading malware. **Malware Distribution:** Injection of code to automatically download and execute malware on users' devices. **Reputation Damage:** Loss of user trust, site traffic, and potential customers due to security risks.	
Suggested Countermeasures	
Input Validation: Validate and sanitize user input to ensure it meets expected formats and limits. **Output Encoding:** Encode user-generated content before displaying it to prevent script execution. **HTTP Only Cookies:** Use HTTP Only flag for cookies to prevent client-side script access. **Content Security Policy (CSP):** Implement CSP to restrict sources of executable scripts.	
References	
https://portswigger.net/web-security/cross-site-scripting	

Proof of Concept

Payload: <script>alert("XSS")</script>0xbt@gmail.com

Encoded:%3C%73%63%72%69%70%74%3E%61%6C%65%72%74%28%22%58%53%53%22%29%3C%2F%73%63%72%69%70%74%3E0xbt@gmail.com

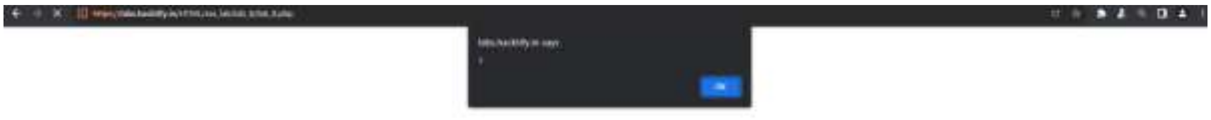


1.8. XSS With File Upload (File Name)

Reference	**Risk Rating**
XSS With File Upload (File Name)	**Low / Medium / High**
Tools Used	
Burp Suite	
Vulnerability Description	
XSS is a web app flaw where attackers inject harmful scripts into pages viewed by users. These scripts steal data, hijack sessions, or change page content. Exploited in input fields, like search boxes, they let attackers run code in victims' browsers.	
How It Was Discovered	
Manual Analysis	
Vulnerable URLs	
https://labs.hacktify.in/HTML/xss_lab/lab_8/lab_8.php	
Consequences of not Fixing the Issue	
Data Theft: Attackers can steal sensitive user information like passwords and credit card details. **Session Hijacking:** They can take control of user sessions, allowing unauthorized actions on behalf of users. **Phishing Attacks:** Creation of convincing phishing links to trick users into revealing credentials or downloading malware. **Malware Distribution:** Injection of code to automatically download and execute malware on users' devices. **Reputation Damage:** Loss of user trust, site traffic, and potential customers due to security risks.	
Suggested Countermeasures	
Input Validation: Validate and sanitize user input to ensure it meets expected formats and limits. **Output Encoding:** Encode user-generated content before displaying it to prevent script execution. **HTTP Only Cookies:** Use HTTP Only flag for cookies to prevent client-side script access. **Content Security Policy (CSP):** Implement CSP to restrict sources of executable scripts.	
References	
https://portswigger.net/web-security/cross-site-scripting	

Proof of Concept

Payload: "><img src=x onerror=alert(1);>"asdf.txt"


labs.hacktify.in says

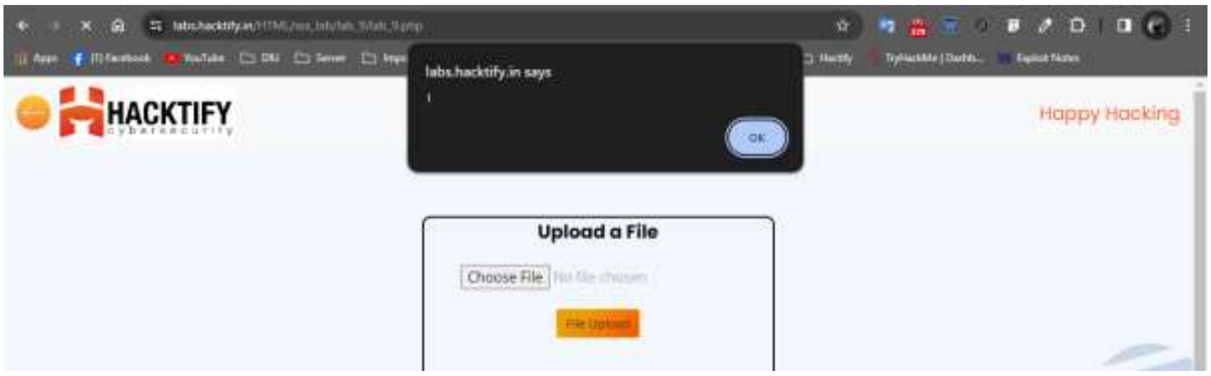
1

1.9. XSS With File Upload (File Content)

Reference	Risk Rating
XSS With File Upload (File Content)	Low / **Medium** / High
Tools Used	
Burp Suite	
Vulnerability Description	
XSS is a web app flaw where attackers inject harmful scripts into pages viewed by users. These scripts steal data, hijack sessions, or change page content. Exploited in input fields, like search boxes, they let attackers run code in victims' browsers.	
How It Was Discovered	
Manual Analysis	
Vulnerable URLs	
https://labs.hacktify.in/HTML/xss_lab/lab_9/lab_9.php	
Consequences of not Fixing the Issue	
Data Theft: Attackers can steal sensitive user information like passwords and credit card details. **Session Hijacking:** They can take control of user sessions, allowing unauthorized actions on behalf of users. **Phishing Attacks:** Creation of convincing phishing links to trick users into revealing credentials or downloading malware. **Malware Distribution:** Injection of code to automatically download and execute malware on users' devices. **Reputation Damage:** Loss of user trust, site traffic, and potential customers due to security risks.	
Suggested Countermeasures	
Input Validation: Validate and sanitize user input to ensure it meets expected formats and limits. **Output Encoding:** Encode user-generated content before displaying it to prevent script execution. **HTTP Only Cookies:** Use HTTP Only flag for cookies to prevent client-side script access. **Content Security Policy (CSP):** Implement CSP to restrict sources of executable scripts.	
References	
https://portswigger.net/web-security/cross-site-scripting	

Proof of Concept

Payload: <script>alert(1)</script>


labs.hacktify.in says

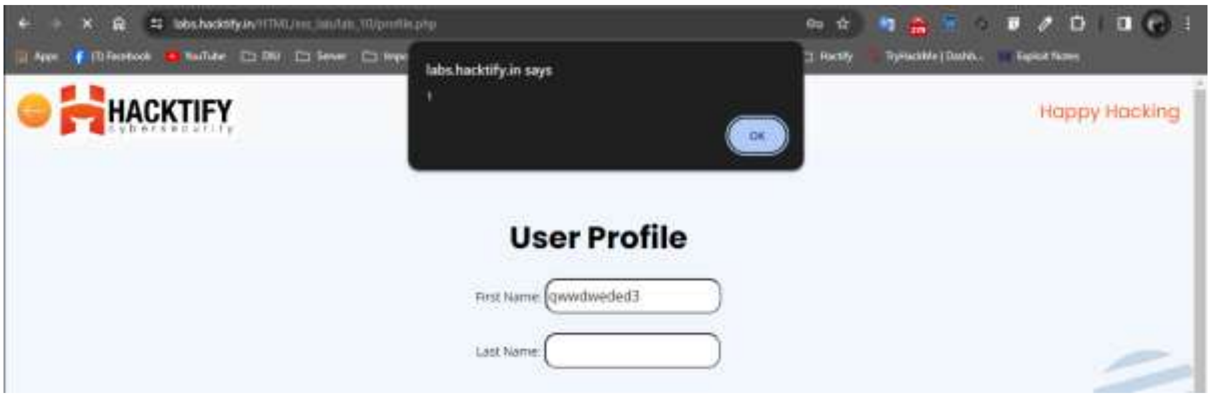
1

1.10. Stored Everywhere!

Reference	**Risk Rating**
Stored Everywhere!	**Low / Medium / High**
Tools Used	
Burp Suite	
Vulnerability Description	
XSS is a web app flaw where attackers inject harmful scripts into pages viewed by users. These scripts steal data, hijack sessions, or change page content. Exploited in input fields, like search boxes, they let attackers run code in victims' browsers.	
How It Was Discovered	
Manual Analysis	
Vulnerable URLs	
https://labs.hacktify.in/HTML/xss_lab/lab_10/lab_10.php	
Consequences of not Fixing the Issue	
Data Theft: Attackers can steal sensitive user information like passwords and credit card details. **Session Hijacking:** They can take control of user sessions, allowing unauthorized actions on behalf of users. **Phishing Attacks:** Creation of convincing phishing links to trick users into revealing credentials or downloading malware. **Malware Distribution:** Injection of code to automatically download and execute malware on users' devices. **Reputation Damage:** Loss of user trust, site traffic, and potential customers due to security risks.	
Suggested Countermeasures	
Input Validation: Validate and sanitize user input to ensure it meets expected formats and limits. **Output Encoding:** Encode user-generated content before displaying it to prevent script execution. **HTTP Only Cookies:** Use HTTP Only flag for cookies to prevent client-side script access. **Content Security Policy (CSP):** Implement CSP to restrict sources of executable scripts.	
References	
https://portswigger.net/web-security/cross-site-scripting	

Proof of Concept

Payload: "><script>alert(1)</script>


labs.hacktify.in says
1
OK

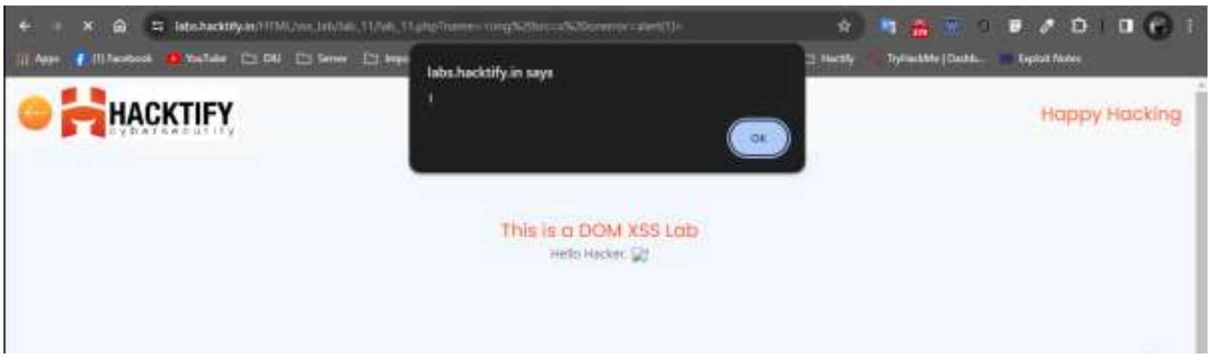
User Profile
First Name: qwwdweded3
Last Name:

1.11. DOM’s Are Love!

Reference	**Risk Rating**
DOM’s Are Love!	**Low / Medium / High**
Tools Used	
Burp Suite	
Vulnerability Description	
XSS is a web app flaw where attackers inject harmful scripts into pages viewed by users. These scripts steal data, hijack sessions, or change page content. Exploited in input fields, like search boxes, they let attackers run code in victims' browsers.	
How It Was Discovered	
Manual Analysis	
Vulnerable URLs	
https://labs.hacktify.in/HTML/xss_lab/lab_11/lab_11.php	
Consequences of not Fixing the Issue	
Data Theft: Attackers can steal sensitive user information like passwords and credit card details. **Session Hijacking:** They can take control of user sessions, allowing unauthorized actions on behalf of users. **Phishing Attacks:** Creation of convincing phishing links to trick users into revealing credentials or downloading malware. **Malware Distribution:** Injection of code to automatically download and execute malware on users' devices. **Reputation Damage:** Loss of user trust, site traffic, and potential customers due to security risks.	
Suggested Countermeasures	
Input Validation: Validate and sanitize user input to ensure it meets expected formats and limits. **Output Encoding:** Encode user-generated content before displaying it to prevent script execution. **HTTP Only Cookies:** Use HTTP Only flag for cookies to prevent client-side script access. **Content Security Policy (CSP):** Implement CSP to restrict sources of executable scripts.	
References	
https://portswigger.net/web-security/cross-site-scripting	

Proof of Concept

Payload: ?name=<img src=x onerror=alert(1)>



labs.hacktify.in says

1

This is a DOM XSS Lab

Hello Hacker.

2. Insecure Direct Object Reference!

2.1. Give Me My Amount!

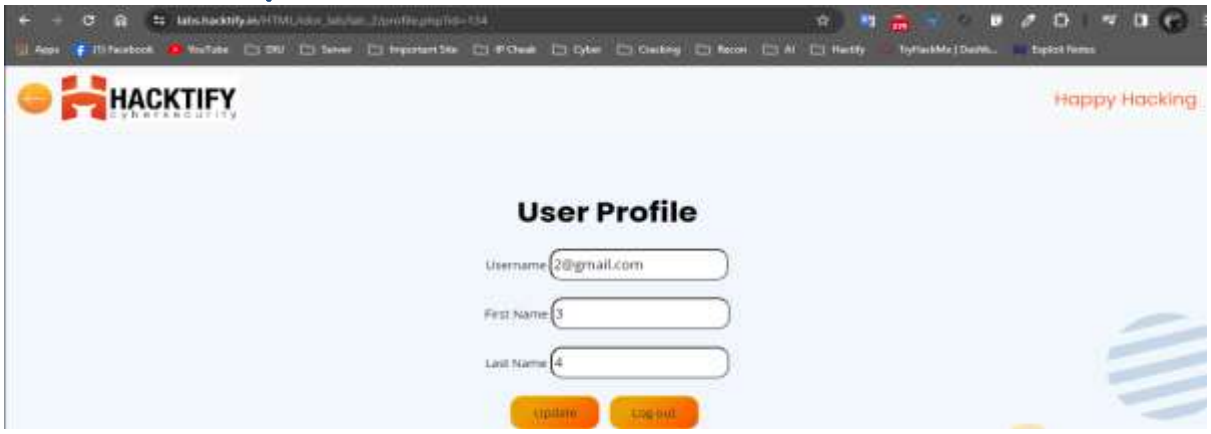
Reference	Risk Rating
Give Me My Amount!	Low / Medium / High
Tools Used	
Burp Suite	
Vulnerability Description	
IDOR is a web application vulnerability where attackers manipulate input to access unauthorized data. To prevent IDOR, developers should implement access controls, strict authentication, and validate user inputs. Regular security audits and testing are crucial.	
How It Was Discovered	
Manual Analysis	
Vulnerable URLs	
https://labs.hacktify.in/HTML/idor_lab/lab_1/lab_1.php	
Consequences of not Fixing the Issue	
Unauthorized Data Access: Attackers can continue to access sensitive information such as personal data, financial records, or proprietary files without permission. **Data Breaches:** The vulnerability leaves the door open for potential data breaches, exposing confidential information to malicious actors. **Legal and Compliance Issues:** Failure to protect user data can lead to legal repercussions, fines, and loss of trust from customers or users. **Financial Loss:** Companies may suffer financial losses due to theft of intellectual property, financial fraud, or loss of business due to reputational damage.	
Suggested Countermeasures	
Implement Strict Access Controls: Ensure that users can only access the data and resources they are authorized to see. **Use Indirect Object References:** Avoid exposing internal object references directly in URLs or forms. Instead, use indirect references that are mapped to actual objects on the server. **Validate User Inputs:** To prevent malicious data manipulation, always validate and sanitize user inputs, and use strong authentication mechanisms like multi-factor authentication to verify user identities. **Enforce Least Privilege:** Follow the principle of least privilege, granting users only the permissions necessary for their roles. **Regular Security Audits:** Conduct frequent security audits and penetration testing to identify and fix	
References	
https://portswigger.net/web-security/access-control/idor	

Proof of Concept

2.2. Stop Polluting My Params!

Reference	**Risk Rating**
Stop Polluting My Params!	**Low / Medium / High**
Tools Used	
Burp Suite	
Vulnerability Description	
IDOR is a web application vulnerability where attackers manipulate input to access unauthorized data. To prevent IDOR, developers should implement access controls, strict authentication, and validate user inputs. Regular security audits and testing are crucial.	
How It Was Discovered	
Manual Analysis	
Vulnerable URLs	
https://labs.hacktify.in/HTML/idor_lab/lab_2/lab_2.php	
Consequences of not Fixing the Issue	
Unauthorized Data Access: Attackers can continue to access sensitive information such as personal data, financial records, or proprietary files without permission. **Data Breaches:** The vulnerability leaves the door open for potential data breaches, exposing confidential information to malicious actors. **Legal and Compliance Issues:** Failure to protect user data can lead to legal repercussions, fines, and loss of trust from customers or users. **Financial Loss:** Companies may suffer financial losses due to theft of intellectual property, financial fraud, or loss of business due to reputational damage.	
Suggested Countermeasures	
Implement Strict Access Controls: Ensure that users can only access the data and resources they are authorized to see. **Use Indirect Object References:** Avoid exposing internal object references directly in URLs or forms. Instead, use indirect references that are mapped to actual objects on the server. **Validate User Inputs:** To prevent malicious data manipulation, always validate and sanitize user inputs, and use strong authentication mechanisms like multi-factor authentication to verify user identities. **Enforce Least Privilege:** Follow the principle of least privilege, granting users only the permissions necessary for their roles. **Regular Security Audits:** Conduct frequent security audits and penetration testing to identify and fix	
References	
https://portswigger.net/web-security/access-control/idor	

Proof of Concept



User Profile

Username 2@gmail.com

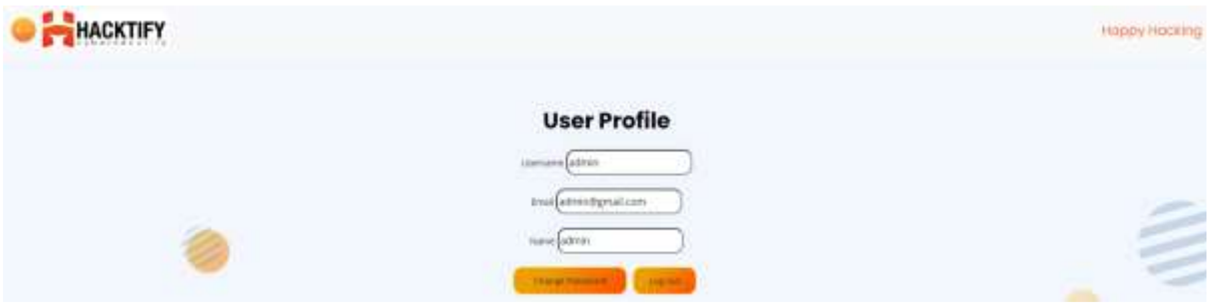
First Name 3

Last Name 4

2.3. Someone Changed My Password!

Reference	Risk Rating
Someone Changed My Password!	**Low / Medium / High**
Tools Used	
Burp Suite	
Vulnerability Description	
IDOR is a web application vulnerability where attackers manipulate input to access unauthorized data. To prevent IDOR, developers should implement access controls, strict authentication, and validate user inputs. Regular security audits and testing are crucial.	
How It Was Discovered	
Manual Analysis	
Vulnerable URLs	
https://labs.hacktify.in/HTML/idor_lab/lab_3/lab_3.php	
Consequences of not Fixing the Issue	
Unauthorized Data Access: Attackers can continue to access sensitive information such as personal data, financial records, or proprietary files without permission. **Data Breaches:** The vulnerability leaves the door open for potential data breaches, exposing confidential information to malicious actors. **Legal and Compliance Issues:** Failure to protect user data can lead to legal repercussions, fines, and loss of trust from customers or users. **Financial Loss:** Companies may suffer financial losses due to theft of intellectual property, financial fraud, or loss of business due to reputational damage.	
Suggested Countermeasures	
Implement Strict Access Controls: Ensure that users can only access the data and resources they are authorized to see. **Use Indirect Object References:** Avoid exposing internal object references directly in URLs or forms. Instead, use indirect references that are mapped to actual objects on the server. **Validate User Inputs:** To prevent malicious data manipulation, always validate and sanitize user inputs, and use strong authentication mechanisms like multi-factor authentication to verify user identities. **Enforce Least Privilege:** Follow the principle of least privilege, granting users only the permissions necessary for their roles. **Regular Security Audits:** Conduct frequent security audits and penetration testing to identify and fix	
References	
https://portswigger.net/web-security/access-control/idor	

Proof of Concept



User Profile

Username: admin

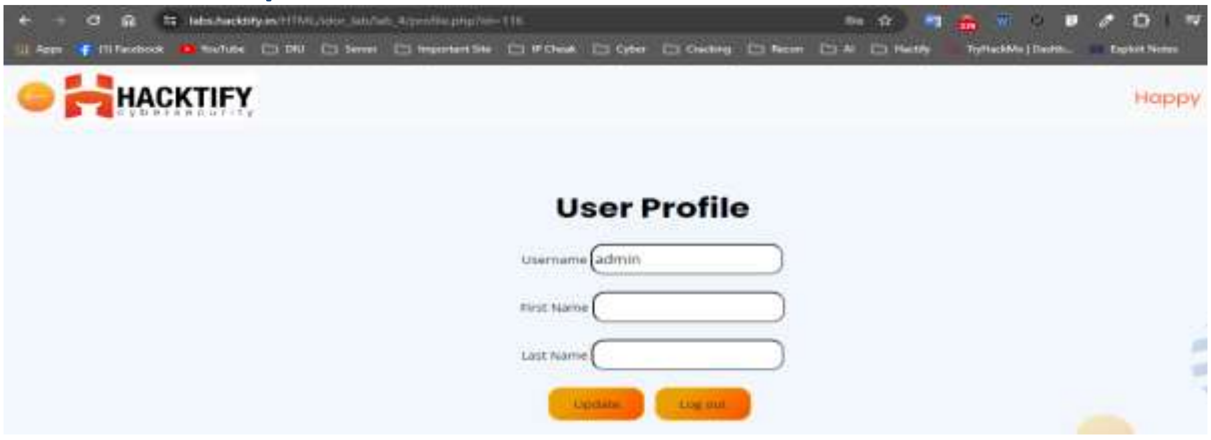
Email: admin@gmail.com

Name: admin

2.4. Change Your Methods!

Reference	Risk Rating
Change Your Methods!	Low / **Medium** / High
Tools Used	
Burp Suite	
Vulnerability Description	
IDOR is a web application vulnerability where attackers manipulate input to access unauthorized data. To prevent IDOR, developers should implement access controls, strict authentication, and validate user inputs. Regular security audits and testing are crucial.	
How It Was Discovered	
Manual Analysis	
Vulnerable URLs	
https://labs.hacktify.in/HTML/idor_lab/lab_4/lab_4.php	
Consequences of not Fixing the Issue	
Unauthorized Data Access: Attackers can continue to access sensitive information such as personal data, financial records, or proprietary files without permission. **Data Breaches:** The vulnerability leaves the door open for potential data breaches, exposing confidential information to malicious actors. **Legal and Compliance Issues:** Failure to protect user data can lead to legal repercussions, fines, and loss of trust from customers or users. **Financial Loss:** Companies may suffer financial losses due to theft of intellectual property, financial fraud, or loss of business due to reputational damage.	
Suggested Countermeasures	
Implement Strict Access Controls: Ensure that users can only access the data and resources they are authorized to see. **Use Indirect Object References:** Avoid exposing internal object references directly in URLs or forms. Instead, use indirect references that are mapped to actual objects on the server. **Validate User Inputs:** To prevent malicious data manipulation, always validate and sanitize user inputs, and use strong authentication mechanisms like multi-factor authentication to verify user identities. **Enforce Least Privilege:** Follow the principle of least privilege, granting users only the permissions necessary for their roles. **Regular Security Audits:** Conduct frequent security audits and penetration testing to identify and fix	
References	
https://portswigger.net/web-security/access-control/idor	

Proof of Concept


labs.hacktify.in/HTML/idor_lab/lab_4/profile.php?id=116

User Profile

Username: admin

First Name:

Last Name: