

# Penetration Testing Report

**Full Name:** Syed Junaid Ahmad Andrabi

**Program:** HCPT

**Date :** 20/07/2024

## 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.

## I. 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.

## II. Scope

The scope of the penetration testing project by Hacktify Cyber Security for clickjacking and HTML injection includes identifying vulnerabilities in the web application's frontend. Testing will focus on detecting clickjacking vulnerabilities that could lead to unauthorized actions by users. HTML injection points will be assessed to identify potential avenues for malicious code insertion. The boundaries of the project exclude testing of backend systems and network infrastructure. Results will be provided with recommendations for mitigation to enhance the application's security posture.

<b>Application , Name</b>	{Lab 1 –Open Redirected} {Lab 2 – HTML Injection}
---------------------------	--

## III. Summary

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

**Total number of Sub-labs: 14(8-Redirected,4-HTML Injection)**

High	Medium	Low
------	--------	-----

4	4	6
---	---	---

- High** - **4 Sub-lab with high difficulty level**
- Medium** - **4 Sub-labs with medium difficulty level**
- Low** - **6 Sub-labs with low difficulty level**

## 1. Open Redirected

### 1.1. A Simple Host!

Reference	Risk Rating
Sub-lab-1: A Simple Host	Low
<b>Tools Used</b>	
Burp-Suite is used to find the vulnerability.	
<b>Vulnerability Description</b>	
Clickjacking, also known as UI Redressing, is a deceptive technique used by attackers to trick users into clicking on hidden or disguised elements on a web page. By overlaying transparent or opaque layers over legitimate content, attackers can manipulate user interactions and perform actions without their knowledge or consent. Clickjacking attacks exploit the inherent trust users place in familiar websites and interfaces to deceive them into unwittingly executing malicious actions. In this lab frame Overlaying is used. =>Embedding legitimate web content within an invisible iframe and positioning malicious elements on top of it to intercept user clicks is known as Frame Overlaying.	
<b>How It Was Discovered</b>	
Automated Tools – Browser Inspect Burp-Suite	
<b>Vulnerable URLs</b>	
<a href="https://labs.hacktify.in/HTML/open_redirect_lab/lab_1/index.php">https://labs.hacktify.in/HTML/open_redirect_lab/lab_1/index.php</a>	
<b>Consequences of not Fixing the Issue</b>	
If this vulnerability is not patched. The user profile will gets deleted in one click, if the user is already logged into the application. (i.e. user has to login again and again when he click something like that).	
<b>Suggested Countermeasures</b>	

# Proof of Concept

**Monitor and Log:** Regularly monitor and log redirect activities to detect and respond to any suspicious behavior.

**Implement Security Headers:** Use security headers like **Content-Security-Policy** (CSP) to restrict the sources that can be loaded.

**Frame Busting Scripts:** Implement JavaScript frame-busting techniques to prevent your site from being loaded in an iframe.

## References

<https://www.coursera.org/>

<https://portswigger.net/web-security/clickjacking>

<https://owasp.org/www-community/attacks/Clickjacking>

<https://www.imperva.com/learn/application-security/clickjacking/>

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

Before starting the lab.

The screenshot shows the Burp Suite Community Edition interface. On the left, the Site map tab is selected, displaying a list of URLs from the target website. On the right, a browser window titled "Hackify Labs" is open, showing a login page with a placeholder "Add the malicious website name to the vulnerable parameter and hit enter". The browser's address bar shows the URL <https://labs.hackify.in>. The browser window has a blue button labeled "Start Lab". Below the browser window, the Burp Suite interface shows the Request and Response panes for a selected item in the site map. The Request pane shows the raw HTTP traffic, and the Response pane shows the HTML response from the server. The response includes standard headers like Content-Type, Content-Length, and Vary, along with a Set-Cookie header for a session ID.

After starting lab(i.e. click on start lab) => It redirects to login page.

Burp Suite Community Edition v2024.5.5 - Temporary Project

Site map Scope Issue definitions

Host Method URL Params Status Code Length MIME type

Request

```
Pretty Raw Hex
1 GET /HTML/open_redirect_lab/lab_1/open_redirect_1.php
2 Host: labs.hacktify.in
3 Cookie: _ga=GA1.1.172121826.1.0.172121826.0.0; _gat=GAI.1.172121826.172121826; cf_clearance=pjZDw5Goc0chJahSDwvLkf_3TQg; PHPSESSID=e931704b72afebf1e6effd738da...
4 Sec-Ch-Ua: "Not(A)Brand";v="0", "Chromium";v="126"
5 Sec-Ch-Ua-Mobile: 10
6 Sec-Ch-Ua-Platform: "Windows"
7 Accept-Language: en-US
8 Upgrade-Insecure-Requests: 1
9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.6479.127 Safari/537.36
10 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,*/*;q=0.8,application/signed-exchange;v=3;q=0.7
11 Sec-Fetch-Site: same-origin
12 Sec-Fetch-Mode: navigate
13 Sec-Fetch-User: 11
14 Sec-Fetch-Dest: document
15 Referer:
```

Response

```
Pretty Raw Hex Render
1 HTTP/2 302 Found
2 Content-Type: text/html; charset=UTF-8
3 Date: Sat, 20 Jul 2024 19:57:00 GMT
4 Cache-Control: no-cache, no-store, must-revalidate
5 Pragma: no-cache
6 Set-Cookie: PHPSESSID=a44bbc9dcd161...
7 Secure
8 Location: https://labs.hacktify.in/login
9 Content-Type: text/html; charset=UTF-8
10 Content-Length: 3173
11 Vary: Accept-Encoding,User-Agent
12 Date: Sun, 21 Jul 2024 18:37:51 GMT
13 Server: LiteSpeed
14 X-Turbo-Charged-By: LiteSpeed
15 <html>
16 <head>
17 <meta charset="UTF-8" />
18 <meta name="viewport" content="width=device-width, initial-scale=1.0" />
19 <meta name="description" content="Shelly - Website" />
20 <meta name="author" content="merkulow" />
21 <meta name="keywords" content="" />
22 <link rel="icon" href="/assets/img/favicon.png" />
23 <link rel="stylesheet" type="text/css" href="../../../../assets/css/animate.css" />
24 </link>
```

Response headers

Burp Suite Community Edition v2024.5.5 - Temporary Project

Target: https://labs.hacktify.in

Request

```
Pretty Raw Hex
1 GET /HTML/open_redirect_lab/lab_1/index.php HTTP/2
2 Host: evil.com?
3 Accept-Encoding: gzip, deflate, br
4 Accept: */
5 Accept-Language: en-US;q=0.9,en;q=0.8
6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.6479.127 Safari/537.36
7 Cache-Control: max-age=0
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
```

Response

```
Pretty Raw Hex Render
1 HTTP/2 404 Not Found
2 Content-Type: text/html
3 Date: Sat, 20 Jul 2024 19:17:22 GMT
4 Server: LiteSpeed
5 X-Turbo-Charged-By: LiteSpeed
6
7
8
9
10
11 <!DOCTYPE html>
12 <html>
13 <head>
14 <meta http-equiv="Content-type" content="text/html; charset=utf-8">
15 <meta http-equiv="Cache-control" content="no-cache">
16 <meta http-equiv="Pragma" content="no-cache">
17 <meta http-equiv="Expires" content="0">
18 <meta name="viewport" content="width=device-width, initial-scale=1.0">
19 <title>404 Not Found</title>
20 <style type="text/css">
21   body {
22     font-family: Arial,Helvetica,sans-serif;
23     font-size:14px;
24     line-height:1.428571429;
25     background-color:#ffffff;
26     color:#F3F3F0;
27     padding:0;
28     margin:0;
29   }
30   section,article{
31     display:block;
32     padding:0;
33     margin:0;
34   }
35   .container{
36     margin-left:auto;
37 }
```

Inspector

- Request attributes
- Request query parameters
- Request body parameters
- Request cookies
- Request headers
- Response headers

Disk: 16.0MB

Memory: 235.2MB

# Proof of Concept

## 1.2. Story Of a Beautiful Header!

Reference	Risk Rating
Sub-lab-2 Story Of a Beautiful Header!	Low
<b>Tools Used</b>	
<b>Burp-Suite, Browser.</b>	
<b>Vulnerability Description</b>	
Clickjacking: Exploring the technique of tricking users into clicking on hidden or disguised elements on a web page, leading to unintended actions or data exposure.	
<b>How It Was Discovered</b>	
Automated Tools – Browser Inspect	
<b>Vulnerable URLs</b>	
<a href="https://labs.hackify.in/HTML/open_redirect_lab/lab_2/index.php">https://labs.hackify.in/HTML/open_redirect_lab/lab_2/index.php</a>	
<b>Consequences of not Fixing the Issue</b>	
1: If users are frequently redirected to malicious sites, they may lose trust in your website, leading to a decline in user engagement and reputation damage. 2: Redirect vulnerabilities can be used to bypass security measures and gain unauthorized access to sensitive data, potentially leading to data breaches.	
<b>Suggested Countermeasures</b>	
Prompt users for confirmation before redirecting them to an external site. <a href="#">This adds an extra layer of security by ensuring users are aware of the redirection.</a>	
<b>References</b>	
<a href="https://portswigger.net/web-security/clickjacking">https://portswigger.net/web-security/clickjacking</a>	
<a href="https://owasp.org/www-security/clickjacking/">https://owasp.org/www-security/clickjacking/</a>	

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

The screenshot shows a Burp Suite interface with the following details:

**Request**

```
POST /open_redirect_lab/lab_2/open_redirect_.php HTTP/2
Host: evilhaxx.com
Cookie: _ga=BCT1450G61.1.172121826.1.0.172121826.0.0.;_gat=GAI.1.1058470122.172121826; PHPSESSID=c780c3bb31f97203b71ce8e474c45f20
Sec-Brand: "Hot(A)Brand";v="0"
Sec-Ch-Ua: "Not(A)Brand";v="0", "Chromium";v="126"
Sec-Ch-Ua-Fingerprint: "Windows"
Accept-Language: en-US
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.6470.127 Safari/127.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://labs.hackify.in/HTML/open_redirect_lab/lab_2/index.php
Accept-Encoding: gzip, deflate, br
Priority: u0, i
19
```

**Response**

```
HTTP/2 404 Not Found
Content-Type: text/html
Vary: Accept-Encoding
Date: Sat, 20 Jul 2024 19:01:37 GMT
Server: LiteSpeed
X-Turbo-Charged-By: LiteSpeed
9
10
11 <!DOCTYPE html>
12 <html>
13 <head>
14 <meta http-equiv="Content-type" content="text/html; charset=utf-8">
15 <meta http-equiv="Cache-control" content="no-cache">
16 <meta http-equiv="Pragma" content="no-cache">
17 <meta http-equiv="Expires" content="0">
18 <meta name="viewport" content="width=device-width, initial-scale=1.0">
19 </head>
20 404 Not Found
</title>
21 <body>
22   font-family: Arial,Helvetica,sans-serif;
23   font-size:14px;
24   line-height:1.428571429;
25   background-color:#ffffff;
26   color:#2ECC71;
27   padding:0;
28   margin:0;
29 }
30 section, footer {
31   display: block;
32   padding: 10px;
33   margin: 0;
34 }
35 .container{
36   margin-left: auto;0 highlights
```

**Inspector**

- Request attributes: 2
- Request query parameters: 0
- Request body parameters: 0
- Request cookies: 3
- Request headers: 21
- Response headers: 5

# Proof of Concept

## 1.3. Sanitize Params!!

Reference	Risk Rating
Sub-lab-3: Sanitize Params!!	Medium
Tools Used	
Burp-Suite is used to find the vulnerability.	
Vulnerability Description	
Manipulating URLs. Inducing the server to make requests to external services without directly observing the responses, making detection more challenging.	
How It Was Discovered	
Automated Tools	
Vulnerable URLs	
<a href="https://labs.hackify.in/HTML/open_redirect_lab/lab_3/open_redirect_3.php">https://labs.hackify.in/HTML/open_redirect_lab/lab_3/open_redirect_3.php</a>	
Consequences of not Fixing the Issue	
Attackers can exploit redirect vulnerabilities to redirect users to malicious sites that mimic legitimate ones, tricking users into revealing sensitive information like passwords and credit card details.	
Suggested Countermeasures	
<ol style="list-style-type: none"><li>1. <b>Use a Whitelist:</b> Implement a whitelist of allowed URLs for redirection. This ensures that only predefined, safe URLs can be used.</li><li>2. <b>Validate and Sanitize Input:</b> Ensure that any input used in redirects is validated and sanitized. Only allow redirects to trusted URLs.</li></ol>	
References	
<a href="https://portswigger.net/web-security/clickjacking">https://portswigger.net/web-security/clickjacking</a>	
<a href="https://owasp.org/www-community/attacks/Clickjacking">https://owasp.org/www-community/attacks/Clickjacking</a>	
<a href="https://www.imperva.com/learn/application-security/clickjacking/">https://www.imperva.com/learn/application-security/clickjacking/</a>	

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

=>It takes both email as well as password without any encryption. (hence can directly access it).

Burp Suite Community Edition v2024.5.5 - Temporary Project

Target: https://labs.hacktify.in | HTTP/2

**Request**

```
Pretty Raw Hex
1 GET /HTML/open_redirect_lab/lab_3/open_redirect_3.php?username=hacktify&password=hacktify!url=open_redirect_3_dashboard.php&login=Login HTTP/1.1
2 Host: labs.hacktify.in
3 Cookie: _ga=GA1.1.1096470122.1712211826.0.0.0; _ga=GAI.1.1096470122.1712211826; cf_clearance=hpqyKnoSHLQ3kxgfhEmGFfylzWNC5PK8ZIC90e.ng0r-17121539187-1.0.1.1-43.UhE8uEVySmrdByGvNvXnDS1FQdwHypq48COP_ykB4Ch05jvB_p120vSGos0chbJabSKbvmpLkf_3T0g; PHPSESSID=b560224b3375865bbcf1c7e96557a1
4 Sec-Ch-Ua: "Not/A)Brand";v="8", "Chromium";v="126"
5 Sec-Ch-Ua-Mobile: ?0
6 Sec-Ch-Ua-Platform: "Windows"
7 Accept-Language: en-US
8 Upgrade-Insecure-Requests: 1
9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.6478.127 Safari/537.36
0 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
1 Sec-Fetch-Site: same-origin
2 Sec-Fetch-Mode: navigate
3 Sec-Fetch-User: ?1
4 Sec-Fetch-Dest: document
5 Referer: https://labs.hacktify.in/HTML/open_redirect_lab/lab_3/open_redirect_3.php
6 Accept-Encoding: gzip, deflate, br
7 Priority: u=0, i
8
9
```

**Response**

```
Pretty Raw Hex Render
1 HTTP/2 302 Found
2 X-Powered-By: PHP/7.4.33
3 Expires: Thu, 19 Nov 1981 08:52:00 GMT
4 Cache-Control: no-cache, no-store, must-revalidate, max-age=0
5 Pragma: no-cache
6 Set-Cookie: PHPSESSID=4701ea88dbfdb83elc67cea0lc103f7b; path=/; secure
7 Location: open_redirect_3_dashboard.php
8 Content-Type: text/html; charset=UTF-8
9 Content-Length: 4339
10 Vary: Accept-Encoding,User-Agent
11 Date: Sun, 21 Jul 2024 21:43:18 GMT
12 Server: LiteSpeed
13 X-Turbo-Charged-By: LiteSpeed
14
15 <html>
16   <head>
17     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
18     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
19     <meta name="Keywords" content="" />
20     <link rel="icon" href="../../assets/img/favicon.png" />
21     <link rel="stylesheet" type="text/css" href="../../assets/css/animate.css" />
22     <link rel="stylesheet" type="text/css" href="../../assets/css/bootstrap.min.css" />
23     <link rel="stylesheet" type="text/css" href="../../assets/css/font-awesome.min.css" />
24     <link rel="stylesheet" type="text/css" href="../../assets/css/main.css" />
25     <link rel="stylesheet" type="text/css" href="../../assets/css/responsive.css" />
26   </head>
27   <body>
28     <div>
29       <h1>Welcome to Hacktify</h1>
30       <p>Please log in to continue.</p>
31       <form action="open_redirect_3_dashboard.php" method="post">
32         <input type="text" name="username" placeholder="Username" />
33         <input type="password" name="password" placeholder="Password" />
34         <input type="submit" value="Log in" />
35       </form>
36     </div>
37   </body>
38 </html>
```

**Inspector**

- Request attributes: 2
- Request query parameters: 4
- Request body parameters: 0
- Request cookies: 4
- Request headers: 22
- Response headers: 12

0 highlights | 4,802 bytes | 345 millis

## Proof of Concept

### 1.4. Patterns are Important!

Reference	Risk Rating
Patterns are Important	Medium
Tools Used	
Burp-Suite is used to find the vulnerability.	
Vulnerability Description	
<b>A redirect vulnerability in a sign-in process occurs when an application improperly handles URL redirections during authentication. This can be exploited by attackers to redirect users to malicious sites, potentially leading to credential theft and other security issues</b>	
How It Was Discovered	
Automated Tools –Burp-suite	
Vulnerable URLs	
<a href="https://labs.hackify.in/HTML/open_redirect_lab/lab_4/open_redirect_4.php">https://labs.hackify.in/HTML/open_redirect_lab/lab_4/open_redirect_4.php</a>	
Consequences of not Fixing the Issue	
Attackers can exploit redirect vulnerabilities to redirect users to malicious sites that mimic legitimate ones, tricking users into revealing sensitive information like passwords and credit card details.	
Suggested Countermeasures	
Ensure that any input used in redirects is validated and sanitized. Only allow redirects to trusted URLs.	
References	
<a href="https://portswigger.net/web-security/clickjacking">https://portswigger.net/web-security/clickjacking</a>	
<a href="https://www.imperva.com/learn/application-security/clickjacking/">https://www.imperva.com/learn/application-security/clickjacking/</a>	

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

```

Request
Pretty Raw Hex
1 GET /HTML/open_redirect_lab/lab_4/open_redirect_4.php?username=hackify&password=hackify!&url=open_redirect_4_dashboard.php&login=Login HTTP/2
2 Host: labs.hackify.in
3 Cookie: _ga_EC17F496TQ=GS1.1.17212C1826.1.0.17212C1826.0.0.0; _gax_ha.1.109447012.1.17212C1826.1.0.17212C1826.0.0.0; _ga=GA.1.109447012.1.17212C1826; cf_clearance=hpqyWno0H12NxgHmGPy1aWC5DQH3ZIC9Qe.ngRx-1721530167-1.0.1.1-43.UxMDuTyIaasclByGwMufS1FCdavHyqpk0f.yh084Ko5jyv_p12Ov9GecochbJmbSHQvmpLkF_.3T0g; PHPSESSID=34bd4fb4430d81e1434Cafefeb59b0f
4 Sec-Ch-Ua: "Not(A)Brand";v="8", "Chromium";v="126"
5 Sec-Ch-Ua-Mobile: ?0
6 Sec-Ch-Ua-Platform: "Windows"
7 Accept-Language: en-US
8 Upgrade-Insecure-Requests: 1
9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.6470.127 Safari/537.36
10 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
11 Sec-Fetch-Site: same-origin
12 Sec-Fetch-Mode: navigate
13 Sec-Fetch-User: ?1
14 Sec-Fetch-Dest: document
15 Referer: https://labs.hackify.in/HTML/open_redirect_lab/lab_4/open_redirect_4.php
16 Accept-Encoding: gzip, deflate, br
17 Priority: u=0, i
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35

```

Response

```

Pretty Raw Hex Render
1 HTTP/2 302 Found
2 X-Powered-By: PHP/7.4.33
3 Expires: Thu, 19 Nov 1901 08:00:00 GMT
4 Cache-Control: no-cache, no-store, must-revalidate, max-age=0
5 Pragma: no-cache
6 Set-Cookie: PHPSESSID=04265ff79da214281a03ad5ac7fcc2ae; path=/; secure
7 Location: open_redirect_4_dashboard.php
8 Content-Type: text/html; charset=UTF-8
9 Content-Length: 4377
10 Vary: Accept-Encoding,User-Agent
11 Date: Mon, 22 Jul 2024 17:27:48 GMT
12 Server: LiteSpeed
13 X-Turbo-Charged-By: LiteSpeed
14
15
16 <html>
17   <head>
18     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
19     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
20     <meta name="keywords" content="" />
21     <link rel="icon" href="../../assets/img/favicon.png" />
22     <link rel="stylesheet" type="text/css" href="../../assets/css/animate.css" />
23     <link rel="stylesheet" type="text/css" href="../../assets/css/bootstrap.min.css" />
24     <link rel="stylesheet" type="text/css" href="../../assets/css/font-awesome.min.css" />
25     <link rel="stylesheet" type="text/css" href="../../assets/css/main.css" />
26     <link rel="stylesheet" type="text/css" href="../../assets/css/responsive.css" />
27   </head>
28   <title>
29     Login
30   </title>
31 
```

## 1.5. File Upload!?Redirect IT!

Reference	Risk Rating
Sub-lab-1: File Uplaod!?Redirect IT!	Low
<b>Tools Used</b>	
Burp-Suite is used to find the vulnerability.	
<b>Vulnerability Description</b>	
Accepts any kind of extension file.	
<b>How It Was Discovered</b>	

## Proof of Concept

Automated Tools – Browser Inspect Burp suite

## Vulnerable URLs

[https://labs.hackify.in/HTML/open\\_redirect\\_lab/lab\\_5/index.php](https://labs.hackify.in/HTML/open_redirect_lab/lab_5/index.php)

## Consequences of not Fixing the Issue

Attackers can upload malicious files, which can then be executed on the server or downloaded by users, spreading malware.

## Suggested Countermeasures

Regularly monitor and log file upload activities to detect and respond to any suspicious behavior

## References

<https://portswigger.net/web-security/clickjacking>

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

Request

Pretty	Raw	Hex
1 GRT /HTML/open_redirect_lab_5/open_redirect_5.php HTTP/2		
2 Host: labs.hacktify.in		
3 Cookie: <code>ga_BU1748GTQ_GSL_1.17121202E_1.0.17121202E_0.0.0; _ga=GAI.1.1056470122.17121202E; cf_clearance=hpqyKnoSHLQ3xrgfRmJPyIaWNC5P0K8Z1C5Q_gNgB-1721530167-1.0.1-i-43.UhPSuUvYsardByGyG; PHPSESSID=6400e70513d42c4580685874af54d1; p120966s=0; cbh=absHdvaplf_F-370g; PHPSESSID=6400e70513d42c4580685874af54d1</code>		
4 Sec-Ch-Ua: "Not(brand);v="8, "Chromium";v="126"		
5 Sec-Ch-Ua-Mobile: ?0		
6 Sec-Ch-Ua-Platform: "Windows"		
7 Accept-Language: en-US		
8 Upgrade-Insecure-Requests: 1		
9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.6478.127 Safari/537.36		
10 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/*,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7		
11 Sec-Fetch-Site: same-origin		
12 Sec-Fetch-Mode: navigate		
13 Sec-Fetch-User: ?1		
14 Sec-Fetch-Dest: document		
15 Referer: https://labs.hacktify.in/HTML/open_redirect_lab_5/index.php		
16 Accept-Encoding: gzip, deflate, br		
17 Priority: u0,i		
18		
19		

Response

Pretty	Raw	Hex	Render
1 HTTP/2 302 Found			
2			X-Powered-By: PHP/7.4.33
3			Expires: Thu, 19 Nov 1981 08:52:00 GMT
4			Cache-Control: no-cache, no-store, must-revalidate, max-age=0
5			Pragma: no-cache
6			Set-Cookie: PHPSESSID=6400e70513d42c4580685874af54d1; path=/; secure
7			Location: https://labs.hacktify.in/.Login/index.php
8			Content-Type: text/html; charset=UTF-8
9			Content-Length: 4034
10			Vary: Accept-Encoding,User-Agent
11			Date: Mon, 22 Jul 2024 17:37:41 GMT
12			Server: LiteSpeed
13			X-Turbo-Charged-By: LiteSpeed
14			
15			<html>
16			<meta charset="UTF-8" />
17			<meta name="viewport" content="width=device-width, initial-scale=1.0" />
18			<meta name="keywords" content="" />
19			<link rel="icon" href="/.assets/img/favicon.png" />
20			<link rel="stylesheet" type="text/css" href="/assets/css/animate.css" />
21			<link rel="stylesheet" type="text/css" href="/assets/css/bootstrap.min.css" />
22			<link rel="stylesheet" type="text/css" href="/.assets/css/bootstrap.min.css" />
23			<link rel="stylesheet" type="text/css" href="/.assets/css/font-awesome.min.css" />
24			<link rel="stylesheet" type="text/css" href="/.assets/css/main.css" />
25			<link rel="stylesheet" type="text/css" href="/.assets/css/responsive.css" />
26			
27			<title>
28			Upload a File
29			</title>
30			<style>
31			
32			
33			
34			
35			

Inspector

Request attributes	2
Request query parameters	0
Request body parameters	0
Request cookies	4
Request headers	22
Response headers	12

The screenshot displays two instances of the Burp Suite interface. The top instance shows a Network tab with a list of captured requests to various domains like 'example.com' and 'iana.org'. A specific request to 'https://labs.hacktify.in' is selected, showing its detailed structure in the Request and Response panes. The bottom instance shows a Repeater tab with a single request to 'http://www.iana.org' being sent. To the right of both instances, a browser window titled 'Example Domains' is open, displaying the 'iana.org' homepage. The page includes sections for 'Domain Names', 'Number Resources', and 'Protocol Assignments', along with links to 'IANA-managed Reserved Domains' and 'Example domains'.

## 1.6. Same Param Twice!

Reference	Risk Rating
Sub-lab-6 Same Param Twice	High
Tools Used	

# Proof of Concept

Burp-Suite
Vulnerability Description
A redirect vulnerability, often referred to as an <b>open redirect</b> , happens when an application allows user-controlled input to determine the URL to which users are redirected without proper validation. This can be exploited to redirect users to malicious sites.
How It Was Discovered
Automated Tools – Burp Suite
Vulnerable URLs
<a href="https://labs.hacktify.in/HTML/open_redirect_lab/lab_6/open_redirect_6.php">https://labs.hacktify.in/HTML/open_redirect_lab/lab_6/open_redirect_6.php</a>
Consequences of not Fixing the Issue
Attackers can use the redirect to capture sensitive data such as login credentials, personal information, or financial details.
Suggested Countermeasures
Avoid using user-controlled input directly in redirects. Instead, use server-side logic to determine the destination.
References
<a href="https://portswigger.net/web-security/clickjacking">https://portswigger.net/web-security/clickjacking</a>

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

The screenshot shows the Burp Suite interface with the following details:

**Request:**

```
GET /HTML/open_redirect_lab/lab_6/open_redirect_6.php?username=hactify&password=hactify123
Host: labs.hacktify.in
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.6478.127 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: 1
Sec-Fetch-Dest: document
Referer: https://labs.hacktify.in/HTML/open_redirect_lab/lab_6/open_redirect_6.php
Accept-Encoding: gzip, deflate, br
Priority: u0, i

```

**Response:**

```
HTTP/2 302 Found
X-Powered-By: PHP/7.4.33
Expires: Mon, 22 Jul 2024 18:31:00 GMT
Cache-Control: no-cache, no-store, must-revalidate, max-age=0
Pragma: no-cache
Set-Cookie: PHPSESSID=19bd0ed7d811138346543ba2cc06db4f0; path=/; secure; HttpOnly; SameSite=None
Content-Type: text/html; charset=UTF-8
Content-Length: 4337
Vary: Accept-Encoding,User-Agent
Date: Mon, 22 Jul 2024 18:31:00 GMT
Server: LiteSpeed
X-Turbo-Charged-By: LiteSpeed

```

**Inspector:**

- Request attributes: 2
- Request query parameters: 4
- Request body parameters: 0
- Request cookies: 4
- Request headers: 22
- Response headers: 12

Bottom status bar: 4,800 bytes | 298 millis

## 1.7. Domains? Not Always!

Reference	Risk Rating
Sub-lab-7: Domain? Not Always	High
<b>Tools Used</b>	
Burp-Suite	
<b>Vulnerability Description</b>	
We can add up the redirected request by adding a url of any choice ,where we want to redirect the user, inside request code we have to add our desired url after authentic url separated by & url=desiredurl example(&url=www.google.com) and it will add the location with authentic redirected page.	
<b>How It Was Discovered</b>	
Automated Tools –Burp Suite (Browser inspect)	
<b>Vulnerable URLs</b>	
<a href="https://labs.hackify.in/HTML/open_redirect_lab/lab_7/open_redirect_7.php">https://labs.hackify.in/HTML/open_redirect_lab/lab_7/open_redirect_7.php</a>	
<b>Consequences of not Fixing the Issue</b>	
Attackers can exploit redirect vulnerabilities to redirect users to malicious sites that mimic legitimate ones, tricking users into revealing sensitive information like passwords and credit card details.	
<b>Suggested Countermeasures</b>	
<i>Avoid allowing user input to determine the destination URL. Instead, use server-side logic to handle redirections.</i>	
<b>References</b>	
<a href="https://portswigger.net/web-security/clickjacking">https://portswigger.net/web-security/clickjacking</a> <a href="https://brightsec.com/blog/open-redirect-vulnerabilities/">https://brightsec.com/blog/open-redirect-vulnerabilities/</a>	

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

# Proof of Concept

The screenshot shows the Burp Suite interface with the following details:

**Request:**

```
1 GET /HTML/open_redirect_lab/lab_7/open_redirect_7.php?username=hacktify&password=hacktify&url=https://www.google.com/login>Login HTTP/2
2 Host: labs.hacktify.in
3 Cookie: _ga=GA1.1.105647012.1721221026.1.0.1721221026.0.0.0; _gat=GAI.1.105647012.1721221026; cf_clearance=hpqyWn0sH1J3RxpNkGPFy1zWHC59OKR0ZIC5Qe.ngpR-1721538167-1.0.1-143.UmK8uEVy5msrdByGwM0rfdS1FCdmwHypqsk80F_yAoB4CRo5jyW8_p1Z0W5GosOchbJmhSK9vmpLhF_.3T0g; PHPSESSID=c00480b1c472a77d5c1af4738c1e4b
4 Sec-Ch-UA: "Not/A/Brand";v="8", "Chromium";v="126"
5 Sec-Ch-UA-Mobile: 70
6 Sec-Ch-UA-Platform: "Windows"
7 Accept-Language: en-US
8 Upgrade-Insecure-Requests: 1
9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.6478.127 Safari/537.36
10 Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
11 Sec-Fetch-Site: same-origin
12 Sec-Fetch-Mode: navigate
13 Sec-Fetch-User: ?1
14 Sec-Fetch-Dest: document
15 Referer: https://labs.hacktify.in/HTML/open_redirect_lab/lab_7/open_redirect_7.php
16 Accept-Encoding: gzip, deflate, br
17 Priority: u=0, i
18
19
```

**Response:**

```
1 HTTP/2 302 Found
2 X-Powered-By: PHP/7.4.33
3 Expires: Thu, 15 Nov 1981 00:52:00 GMT
4 Cache-Control: no-cache, no-store, must-revalidate, max-age=0
5 Pragma: no-cache
6 Set-Cookie: PHPSESSID=b4b60hdia0fec8a0c5053098c113fb82; path=/; secure
7 Location: ../../Login/index.php
8 Content-Type: text/html; charset=UTF-8
9 Content-Length: 4337
10 Vary: Accept-Encoding,User-Agent
11 Date: Mon, 22 Jul 2024 18:42:46 GMT
12 Server: LiteSpeed
13 X-Turbo-Charged-By: LiteSpeed
14
15 <html>
16   <head>
17     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
18     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
19     <meta name="Keywords" content="" />
20     <link rel="icon" href="../../assets/img/favicon.png" />
21     <link rel="stylesheet" type="text/css" href="../../assets/css/animate.css" />
22     <link
23       rel="stylesheet"
24       type="text/css"
25       href="../../assets/css/bootstrap.min.css"
26     />
27   </head>
28   <link
29     rel="stylesheet"
30     type="text/css"
31     href="../../assets/css/font-awesome.min.css"
32   />
33   <link rel="stylesheet" type="text/css" href="../../assets/css/main.css" />
34   <link rel="stylesheet" type="text/css" href="../../assets/css/responsive.css" />
35 <title>
  Login
</title>
```

**Inspector:** Request attributes: 2, Request query parameters: 4, Request body parameters: 0, Request cookies: 4, Request headers: 22, Response headers: 12.

**Notes:** None.

## 1.8. Style Digit Symbols<3

Reference

Risk Rating

Sub-lab-8: Style Digit Symbols	High
<b>Tools Used</b>	
Burp-Suite	
<b>Vulnerability Description</b>	
We can add up the redirected request by adding an IP of any choice, where we want to redirect the user, inside request code we have to add our desired IP in url section like &url=IP Address and it will add the Location Of IP Address in the url and change the redirected request.	
<b>How It Was Discovered</b>	
Automated Tools –Burp Suite (Browser inspect)	
<b>Vulnerable URLs</b>	
<a href="https://labs.hackify.in/HTML/open_redirect_lab/lab_8/open_redirect_8.php">https://labs.hackify.in/HTML/open_redirect_lab/lab_8/open_redirect_8.php</a>	
<b>Consequences of not Fixing the Issue</b>	
Attackers can exploit redirect vulnerabilities to redirect users to malicious sites that mimic legitimate ones, tricking users into revealing sensitive information like passwords and credit card details.	
<b>Suggested Countermeasures</b>	
<p><b>Where possible, use relative URLs instead of absolute URLs to prevent redirection to external sites.</b></p> <p><b>Use security headers like Content-Security-Policy (CSP) to restrict the sources that can be loaded. This helps prevent unauthorized redirections</b></p>	
<b>References</b>	
<a href="https://portswigger.net/web-security/clickjacking">https://portswigger.net/web-security/clickjacking</a> <a href="https://dzone.com/articles/what-is-an-open-redirection-vulnerability-and-how">https://dzone.com/articles/what-is-an-open-redirection-vulnerability-and-how</a>	

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

# Proof of Concept

Burp Suite Community Edition v2024.5.5 - Temporary Project

Target: https://labs.hackify.in HTTP/2 (140.86.14.128:443)

Request

```
Pretty Raw Hex
1 GET /HTML/open_redirect_lab/lab_0/open_redirect_0.php?username=hackify&password=hackify!url=evil.com>Login HTTP/2
2 Host: labs.hackify.in
3 Cookie: _ga=GA1.1.1096470122.1712221826.1.0.1721221826.0.0; _ga=GAI.1.1096470122.1712221826; cf_clearance=hpqyKno6H1Q3kxgFmF9y1wNC5P0REZIC1C9e.ng0X-1721538167-1.0.1.1-43.UhK8uEVy5msrdYgVanXuDS1f7Gdwvtypgk80P_ykAB4Ce05jvR_p1ZUwSGos0chbJahSKbvnpLhF_3T0g; PHPSESSID=db8c0c5df7b302810660d4d4459a8d
4 Sec-Ch-Ua: "(Not/A)Brand";v="0", "Chromium";v="106"
5 Sec-Ch-Ua-Mobile: ?0
6 Sec-Ch-Ua-Platform: "Windows"
7 Accept-Language: en-US
8 Upgrade-Insecure-Requests: 1
9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.6478.127 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://labs.hackify.in/HTML/open_redirect_lab/lab_0/open_redirect_0.php
Accept-Encoding: gzip, deflate, br
Priority: u=0, i
.
```

Response

```
Pretty Raw Hex Render
1 HTTP/2 302 Found
2 X-Powered-By: PHP/7.4.33
3 Expires: Thu, 19 Nov 1981 08:52:00 GMT
4 Cache-Control: no-cache, no-store, must-revalidate, max-age=0
5 Pragma: no-cache
6 Set-Cookie: PHPSESSID=6d34c37fe7581a6992101ba800674f0t; path=/; secure
7 Location: open_redirect_0_dashboard.php
8 Content-Type: text/html; charset=UTF-8
9 Content-length: 437
10 Vary: Accept-Encoding,User-Agent
11 Date: Mon, 22 Jul 2024 19:12:19 GMT
12 Server: LiteSpeed
13 X-Turbo-Charged-By: LiteSpeed
14
15
16 <html>
17   <head>
18     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
19     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
20     <meta name="keywords" content="" />
21     <link rel="icon" href="../../assets/img/favicon.png" />
22     <link rel="stylesheet" type="text/css" href="../../assets/css/animate.css" />
23     <link
24       rel="stylesheet"
25       type="text/css"
26       href="../../assets/css/bootstrap.min.css"
27     />
28     <link
29       rel="stylesheet"
30       type="text/css"
31       href="../../assets/css/font-awesome.min.css"
32     />
33     <link rel="stylesheet" type="text/css" href="../../assets/css/main.css" />
34     <link rel="stylesheet" type="text/css" href="../../assets/css/responsive.css"
35     />
36   <ttitle>
37     Login
38   </ttitle>
39 </head>
```

Inspector

Selected text: &url=evil.com

Decoded from: URL encoding ( &url=evil.com )

Cancel Apply changes

Request attributes: 2

Request query parameters: 4

Request body parameters: 0

Request cookies: 4

Request headers: 22

Response headers: 12

Done 4,800 bytes | 393 mil

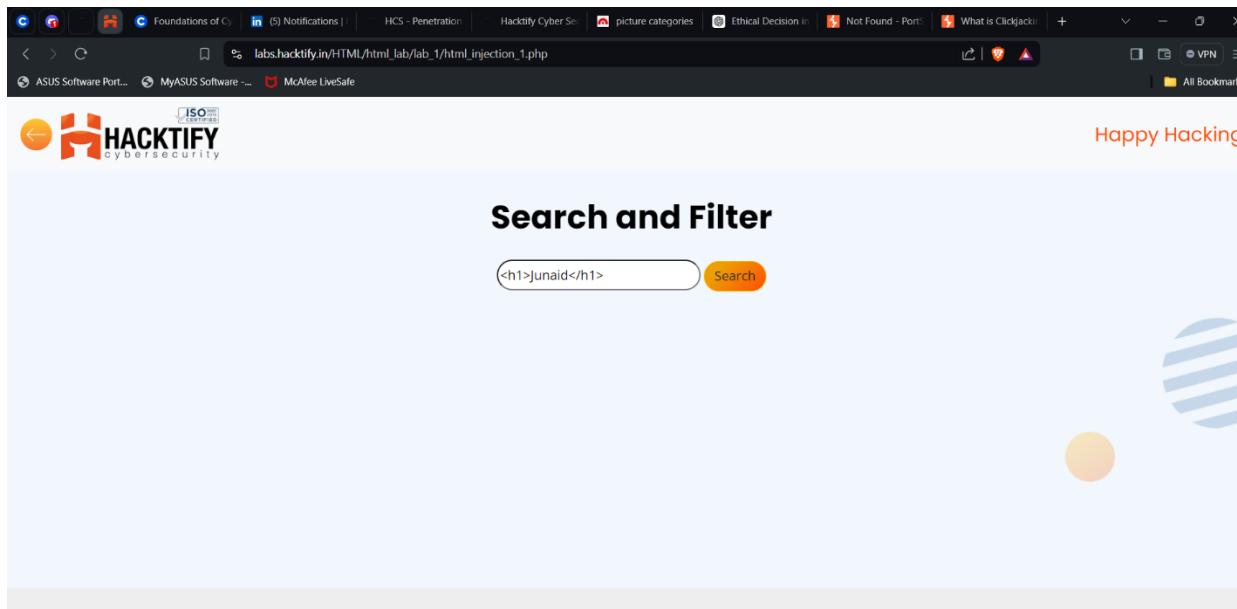
### **3. HTML Injection**

#### **2.1. HTML's are easy!**

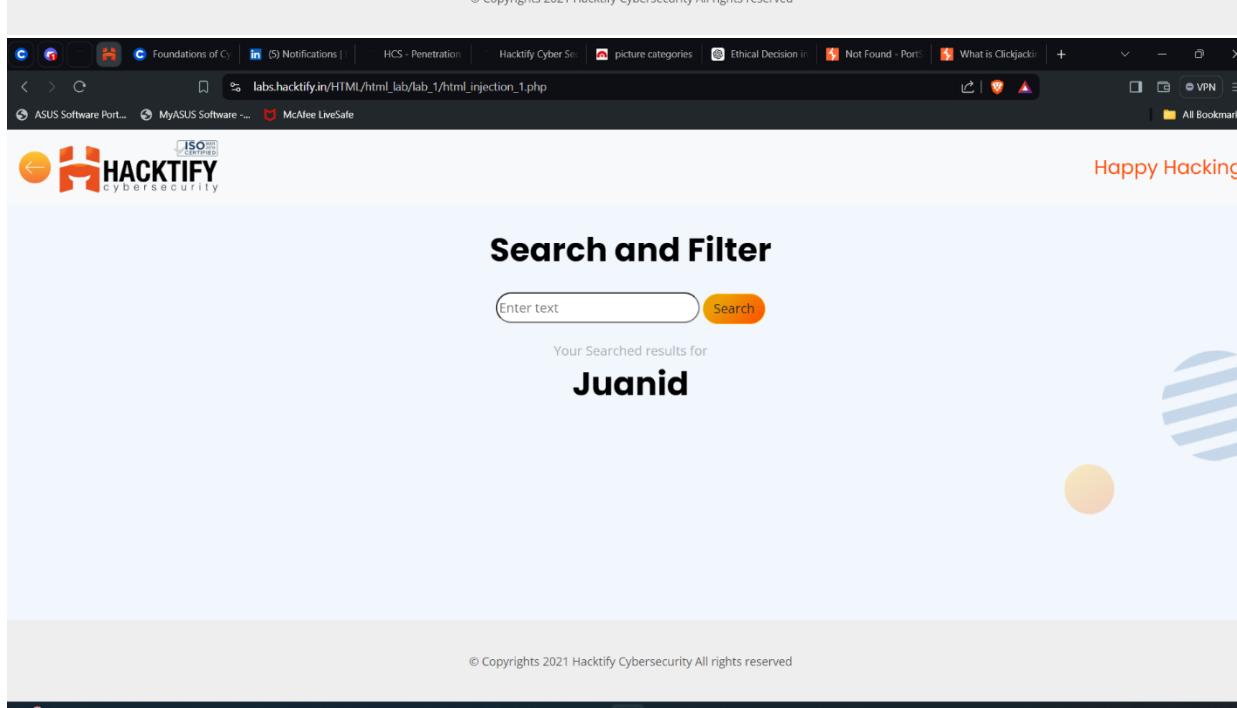
Reference	Risk Rating
Sub-lab-1: HTML's are easy!	Low
<b>Tools Used</b>	
Browser " <u>View Page Sources</u> " is used to find the vulnerability.(Also inspect)	
<b>Vulnerability Description</b>	
The attacker finds an input point in the web application where they can insert HTML code.	
<b>How It Was Discovered</b>	
Automated Tools – Browser View Page Sources (Ctrl + U)	
<b>Vulnerable URLs</b>	
<a href="https://labs.hackify.in/HTML/html_lab/lab_1/index.php">https://labs.hackify.in/HTML/html_lab/lab_1/index.php</a>	
<b>Consequences of not Fixing the Issue</b>	
Malicious HTML can be used to steal session cookies, allowing attackers to impersonate users and gain unauthorized access to their accounts.	
<b>Suggested Countermeasures</b>	
Restrict the types of input that users can provide, especially in fields that will be rendered as HTML. Use dropdowns, radio buttons, and other controlled input methods where possible.	
<b>References</b>	
<a href="https://trainings.internshala.com/blog/html-injections/">https://trainings.internshala.com/blog/html-injections/</a>	
<a href="https://owasp.org/www-community/Injection_Information">https://owasp.org/www-community/Injection_Information</a>	
<a href="https://en.wikipedia.org/wiki/HTML_injection">https://en.wikipedia.org/wiki/HTML_injection</a>	

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

## Proof of Concept

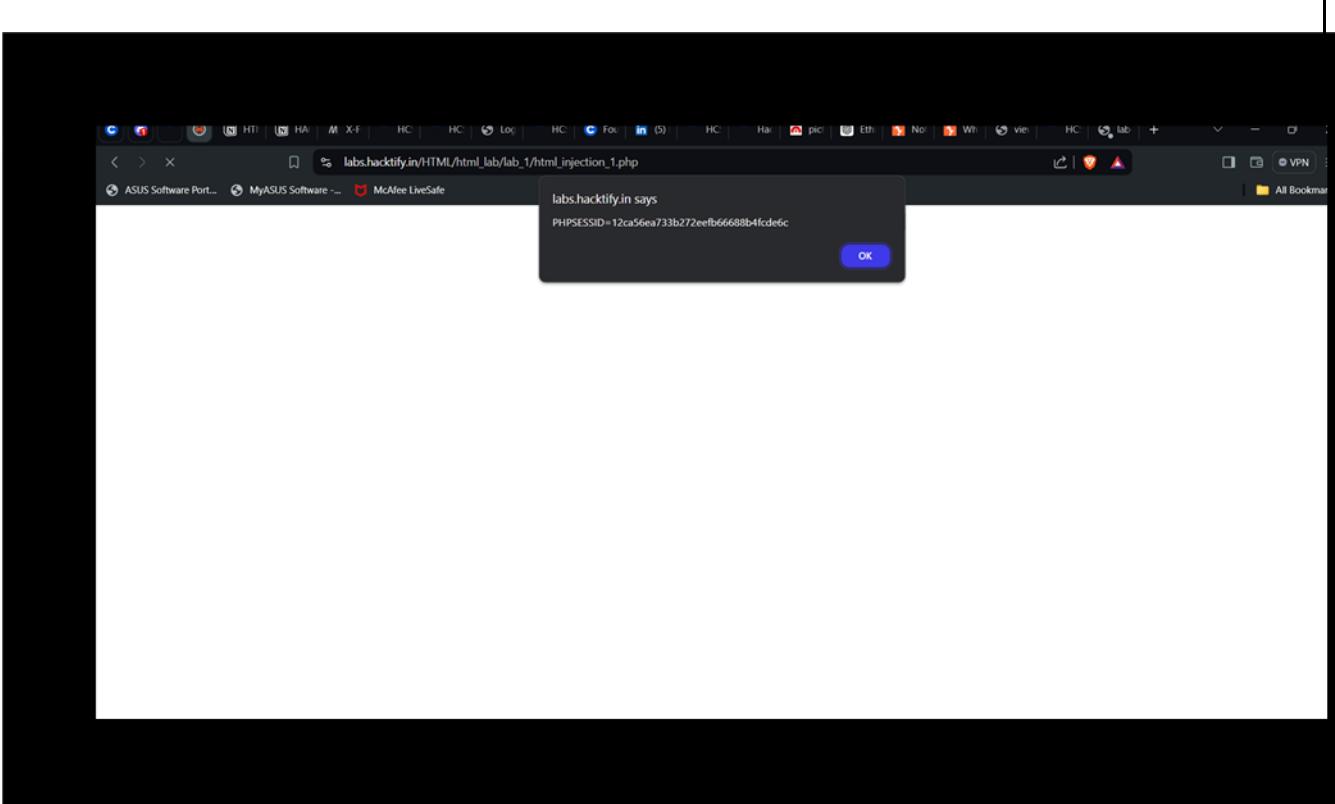
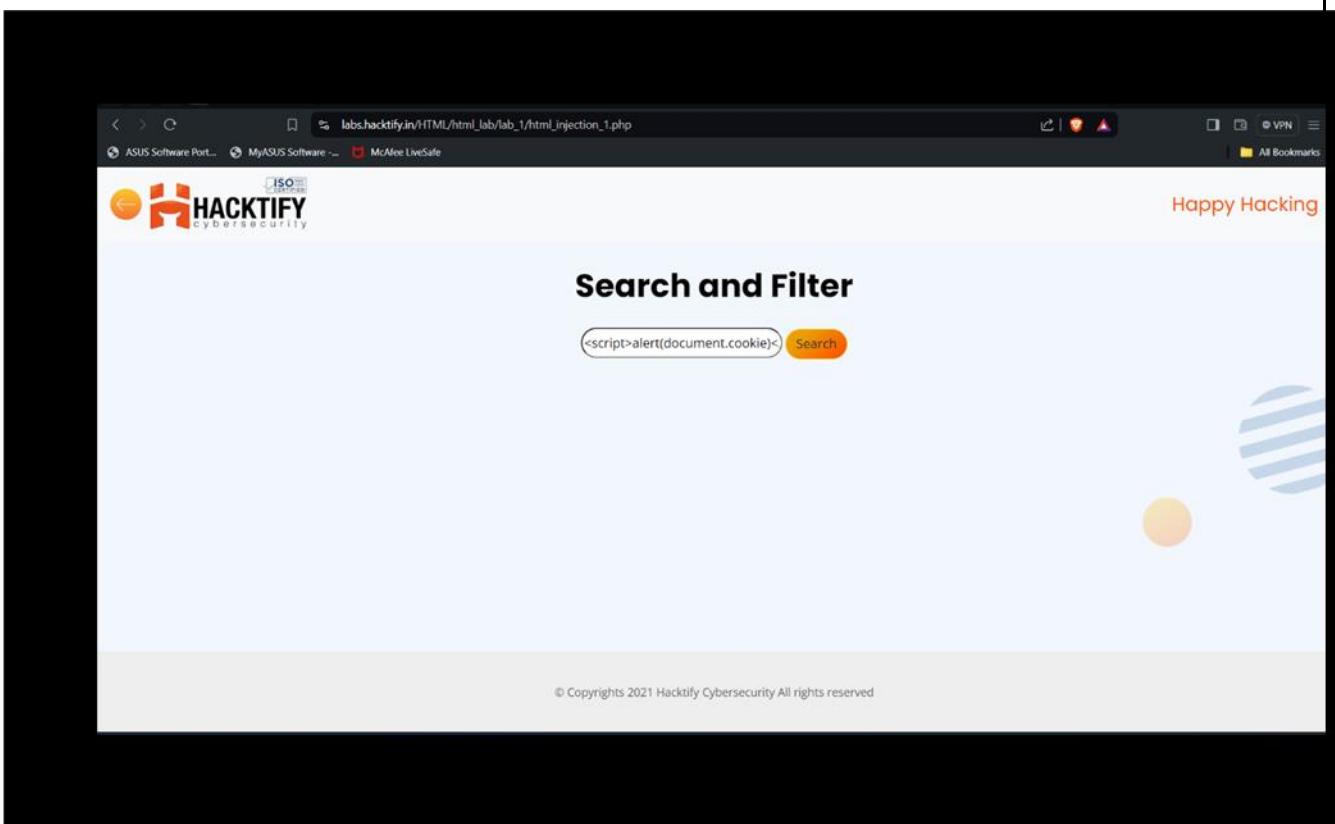


The screenshot shows a web browser window with the URL `labs.hacktify.in/HTML/html_lab/lab_1/html_injection_1.php`. The page title is "Search and Filter". A search bar contains the text `<h1>Junaid</h1>`. The search button is orange with the word "Search". Below the search bar, there is a message: "Your Searched results for **Juanid**". The footer of the page says "© Copyrights 2021 Hacktify Cybersecurity All rights reserved".



This screenshot is identical to the one above, except the search bar now contains the placeholder text "Enter text". The rest of the page, including the search results and footer, remains the same.

=>



## Proof of Concept

### 2.2. Let me Store them!

Reference	Risk Rating
Sub-lab-1: Let me Store them	Low
<b>Tools Used</b>	
Browser " <a href="#">View Page Sources</a> " is used to find the vulnerability.	
<b>Vulnerability Description</b>	
Automated Tools – Browser View Page Sources (Ctrl + U)	
<b>Vulnerable URLs</b>	
<a href="https://labs.hacktify.in/HTML/html_lab/lab_2/index.php">https://labs.hacktify.in/HTML/html_lab/lab_2/index.php</a>	
<b>Consequences of not Fixing the Issue</b>	
Attackers can alter the appearance of your website, damaging your brand's reputation and causing users to lose trust in your site.	
<b>Suggested Countermeasures</b>	
<b>Input Validation and Sanitization:</b> Rigorously validate and sanitize all user inputs to ensure they adhere to expected formats. This helps prevent malicious HTML from being processed	
<b>References</b>	
<a href="https://owasp.org/www-community/Injection_Information">https://owasp.org/www-community/Injection_Information</a>	
<a href="https://en.wikipedia.org/wiki/HTML_injection">https://en.wikipedia.org/wiki/HTML_injection</a>	

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

Happy Hacking

## Register

First Name

Last Name:

Email

Password:

Confirm Password

[Register](#) [Login](#)

© Copyrights 2021 Hacktify Cybersecurity All rights reserved

Happy Hacking

## User Profile

First Name:

Last Name:

Email:

Password

Confirm Password

[Update](#) [Log out](#)

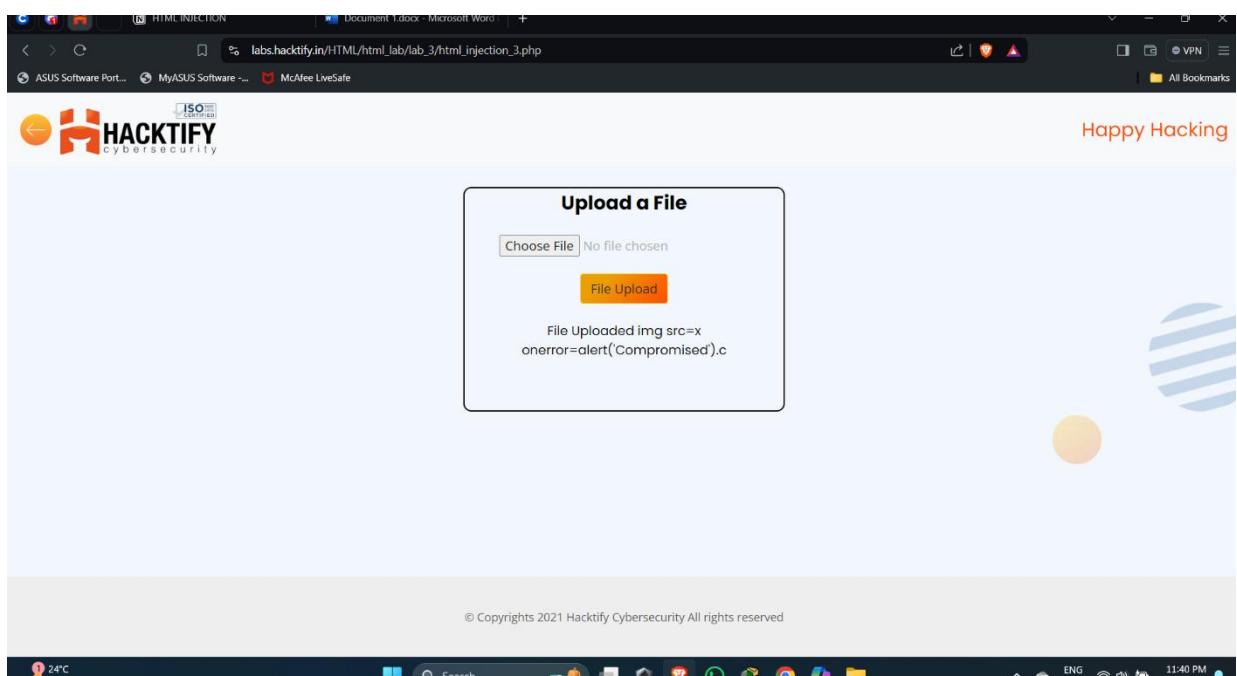
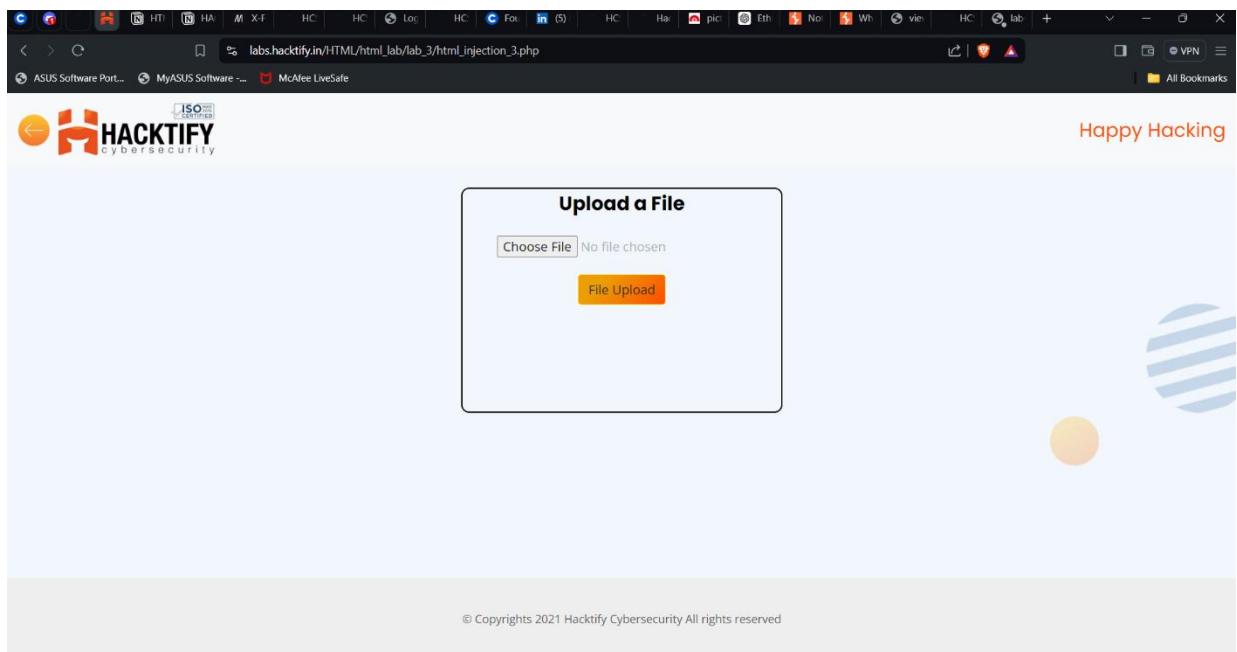
© Copyrights 2021 Hacktify Cybersecurity All rights reserved

## Proof of Concept

### 2.3. Files Names are also vulnerable!

Reference	Risk Rating
Sub-lab-1: Files Names are also vulnerable	Low
<b>Tools Used</b>	
Burp-Suite	
<b>Vulnerability Description</b>	
Allowing attackers to upload files containing malicious HTML code. This can lead to various security issues, including data breaches, malware distribution, and server compromise	
<b>How It Was Discovered</b>	
Automated Tools –Burp-suite	
<b>Vulnerable URLs</b>	
<a href="https://labs.hackify.in/HTML/html_lab/lab_3/index.php">https://labs.hackify.in/HTML/html_lab/lab_3/index.php</a>	
<b>Consequences of not Fixing the Issue</b>	
Uploaded files can exploit server vulnerabilities, allowing attackers to take control of the server. This can result in unauthorized access, data manipulation, and further attacks	
<b>Suggested Countermeasures</b>	
<b>Validate File Types:</b> Ensure that only specific, allowed file types can be uploaded. Verify the file type by checking the file's content, not just its extension .	
<b>Use Antivirus Scanning:</b> Scan all uploaded files for malware using multiple antivirus engines to ensure they are safe.	
<b>References</b>	
<a href="https://owasp.org/www-community/Injection_Information">https://owasp.org/www-community/Injection_Information</a>	
<a href="https://en.wikipedia.org/wiki/HTML_injection">https://en.wikipedia.org/wiki/HTML_injection</a>	
<a href="https://www.wallarm.com/what/html-injection">https://www.wallarm.com/what/html-injection</a>	
<a href="https://www.opswat.com/blog/file-upload-protection-best-practices">https://www.opswat.com/blog/file-upload-protection-best-practices</a>	

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



24°C  
Dover

Search

ENG n1 WiFi 11:40 PM 7/16/2024

# Proof of Concept

The screenshot shows the Burp Suite interface with the 'Proxy' tab selected. A request to [https://labs.hackify.in/HTML/html\\_lab/lab\\_3/html\\_injection\\_3.php](https://labs.hackify.in/HTML/html_lab/lab_3/html_injection_3.php) is being viewed. The request body contains a multipart form-data payload with a file named 'image'. The 'Inspector' tool is open, showing the selected text and its decoded form.

```
POST /HTML/html_lab/lab_3/html_injection_3.php HTTP/1.1
Host: labs.hackify.in
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/106.0.4768.127 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=1
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://labs.hackify.in/HTML/html_lab/lab_3/html_injection_3.php
Content-Type: application/vnd.openxmlformats-officedocument.wordprocessingml.document
Content-Disposition: form-data; name="image"; filename="file compromised.bmp"
Content-Transfer-Encoding: binary
-----WebKitFormBoundaryHWF5ZV0tLNcG6MB5
Content-Disposition: form-data; name="upload"
-----WebKitFormBoundaryHWF5ZV0tLNcG6MB5--
```

The 'Selected text' pane shows the following code:

```
; name="image"; filename="file compromised<br>
```

The 'Decoded from' pane shows the original code:

```
; name="image"; filename="file compromised<br>
```

The 'Inspector' sidebar lists various request attributes and headers.

The screenshot shows a file upload interface on the Hackify website. The URL is [https://labs.hackify.in/HTML/html\\_lab/lab\\_3/html\\_injection\\_3.php](https://labs.hackify.in/HTML/html_lab/lab_3/html_injection_3.php). The 'Choose File' input field has a file selected, and the 'File Upload' button is visible. Below the button, a message indicates the file was uploaded successfully.

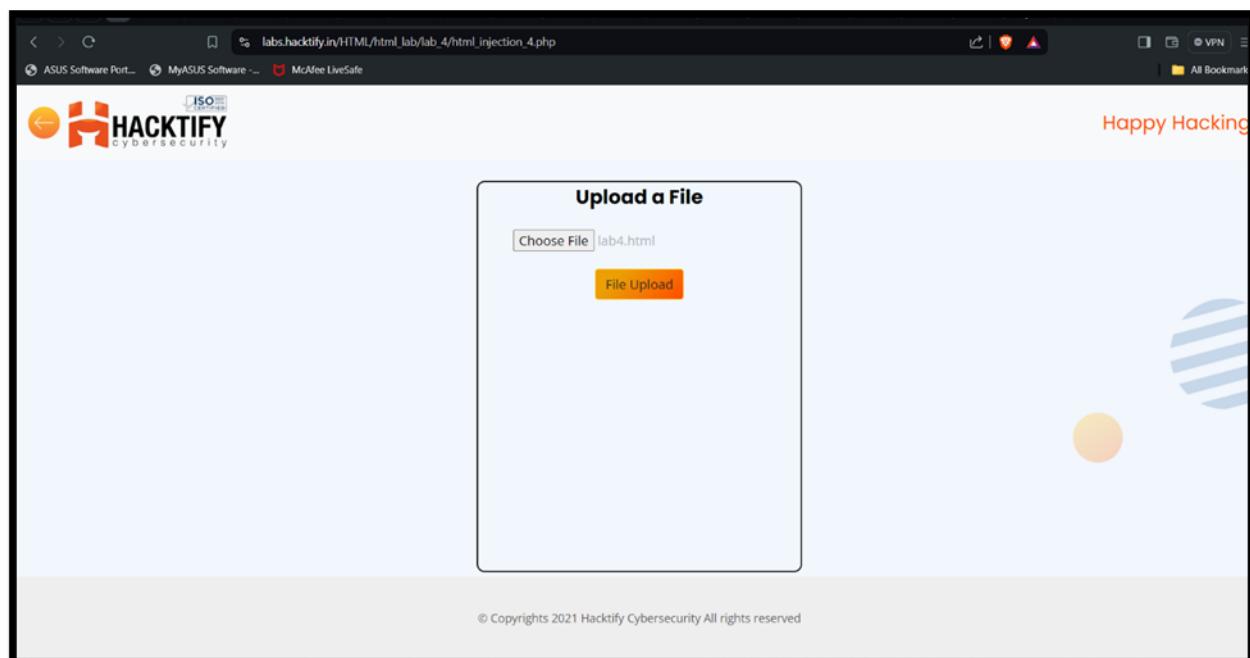
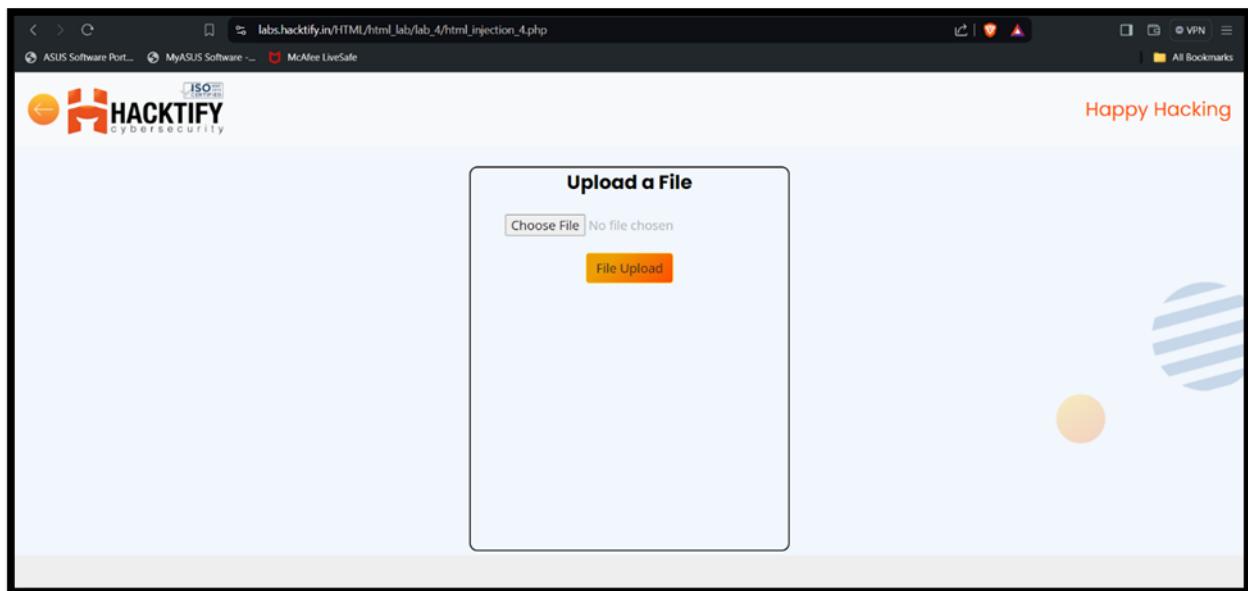
© Copyrights 2021 Hackify Cybersecurity All rights reserved

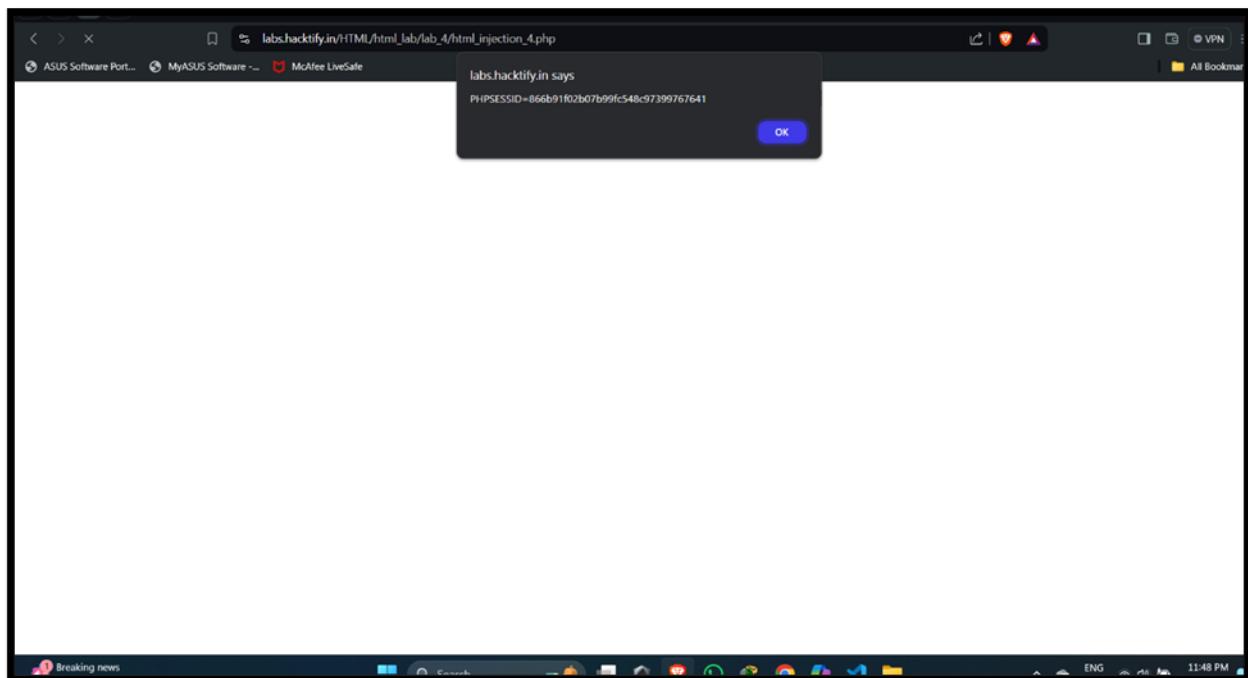
## 2.4. File Content and HTML Injection a perfect pair!

Reference	Risk Rating
Sub-lab-4: File Content and HTML Injection a perfect pair	Medium
<b>Tools Used</b>	
Browser " <u><a href="#">View Page Sources</a></u> " is used to find the vulnerability.	
<b>Vulnerability Description</b>	
The application fails to properly validate the file type, size, or content.	
<b>How It Was Discovered</b>	
Automated Tools – Browser View Page Sources (Ctrl + U)	
<b>Vulnerable URLs</b>	
<u><a href="https://labs.hackify.in/HTML/html_lab/lab_4/index.php">https://labs.hackify.in/HTML/html_lab/lab_4/index.php</a></u>	
<b>Consequences of not Fixing the Issue</b>	
Attackers can steal sensitive information by tricking users into entering data into manipulated forms.	
<b>Suggested Countermeasures</b>	
<b>Set File Size Limits:</b> Restrict the size of uploaded files to prevent denial-of-service (DoS) attacks that exploit large file uploads	
<b>Store Files Securely:</b> Store uploaded files outside the webroot or on a different server to minimize the risk of direct access and execution.	
<b>References</b>	
<u><a href="https://owasp.org/www-community/Injection_Information">https://owasp.org/www-community/Injection_Information</a></u>	
<u><a href="https://en.wikipedia.org/wiki/HTML_injection">https://en.wikipedia.org/wiki/HTML_injection</a></u>	
<u><a href="https://www.acunetix.com/">https://www.acunetix.com/</a></u>	

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

## Proof of Concept





## 2.5. Injecting HTML using URL

Reference	Risk Rating
Sub-lab-5: Injecting HTML using URL	Medium
Tools Used	
Browser " <a href="#">View Page Sources</a> " is used .	
Vulnerability Description	
The attacker finds an input point in the web application where they can insert HTML code.	
How It Was Discovered	
Automated Tools – Browser View Page Sources (Ctrl + U) /Browser Inspector	
Vulnerable URLs	
<a href="https://labs.hackify.in/HTML/html_lab/lab_5/index.php">https://labs.hackify.in/HTML/html_lab/lab_5/index.php</a>	

# Proof of Concept

## Consequences of not Fixing the Issue

Malicious HTML can be used to steal session cookies, allowing attackers to impersonate users.

## Suggested Countermeasures

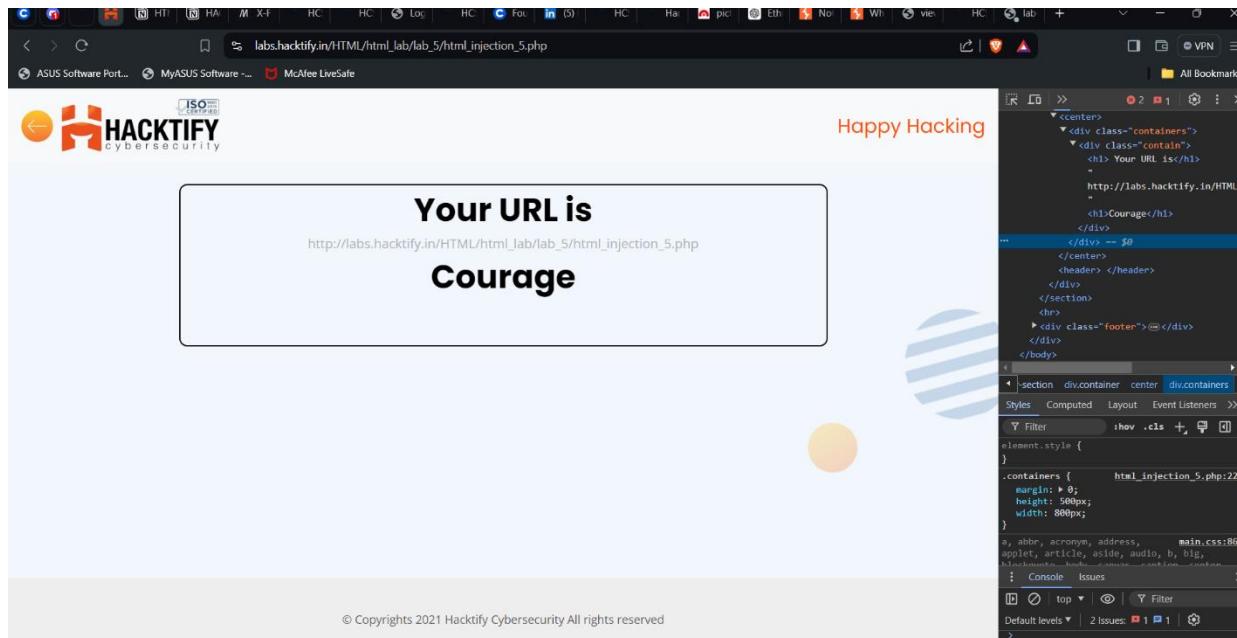
- Validate and sanitize all user inputs.
- Use secure coding practices.
- Implement Content Security Policy (CSP) headers.

## References

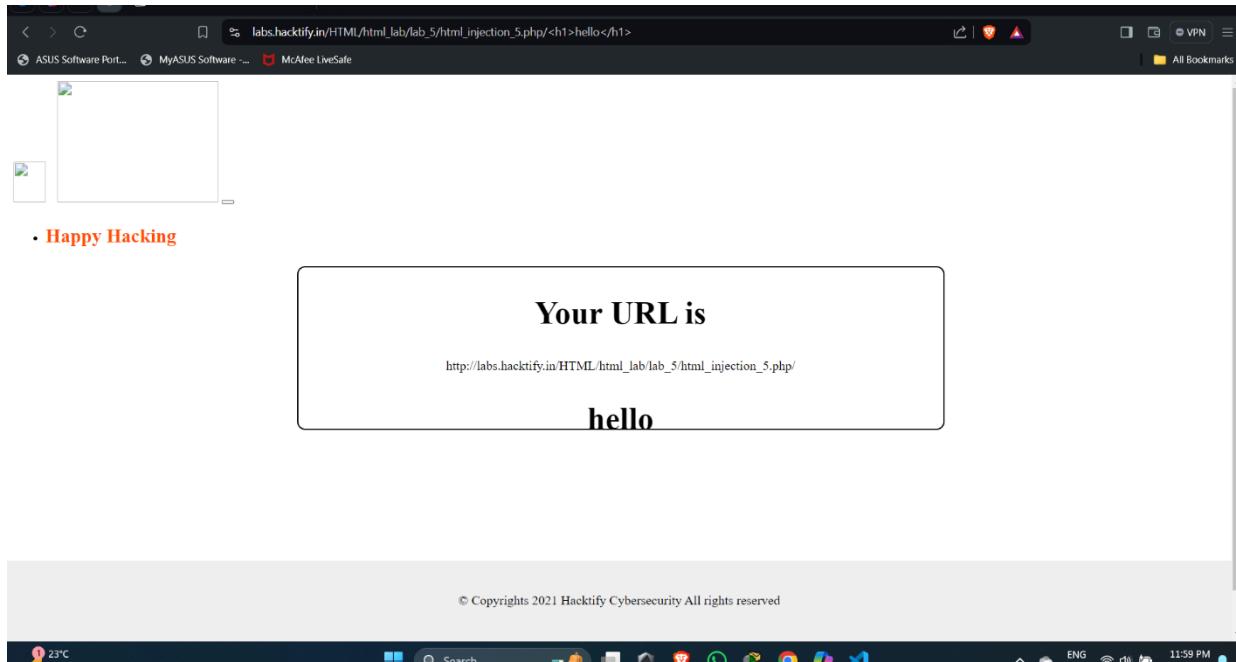
[https://owasp.org/www-community/Injection\\_Information](https://owasp.org/www-community/Injection_Information)

[https://en.wikipedia.org/wiki/HTML\\_injection](https://en.wikipedia.org/wiki/HTML_injection)

<https://www.imperva.com/learn/application-security/html-injection/>



⇒ Another one



## 2.6. Encode It!

Reference	Risk Rating
Sub-lab-6: Encode It	High
<strong>Tools Used</strong>	
Burp Suite	
<strong>Vulnerability Description</strong>	
The attacker finds an input point in the web application where they can insert HTML code. And he can also insert encoded html tags	
<strong>How It Was Discovered</strong>	
Automated Tool----Burp-suite	
<strong>Vulnerable URLs</strong>	
<a href="https://labs.hackify.in/HTML/html_lab/lab_6/index.php">https://labs.hackify.in/HTML/html_lab/lab_6/index.php</a>	
<strong>Consequences of not Fixing the Issue</strong>	
Malicious HTML can be used to steal session cookies, allowing attackers to impersonate users. Attackers can alter the appearance of the website, damaging its reputation.	
<strong>Suggested Countermeasures</strong>	

# Proof of Concept

- Input Validation and Sanitization:** Rigorously validate and sanitize all user inputs to ensure they adhere to expected formats. This helps prevent malicious HTML from being processed.
- Content Security Policy (CSP):** Implement a strong CSP to control the sources from which content can be loaded. This helps prevent the execution of malicious scripts

## References

[https://owasp.org/www-community/Injection\\_Information](https://owasp.org/www-community/Injection_Information)

[https://en.wikipedia.org/wiki/HTML\\_injection](https://en.wikipedia.org/wiki/HTML_injection)

<https://www.acunetix.com/>

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

The screenshot displays the Burp Suite interface and a web browser window side-by-side. On the left, the Burp Suite proxy tab shows a captured POST request to `https://labs.hackify.in/HTML/html_lab_6/html_injection_6.php`. The request body is set to `<h1>courage</h1>`. On the right, a browser window shows the response from the same URL, displaying the HTML content `<h1>courage</h1>`. The browser window includes the Hackify Cybersecurity logo and navigation controls.

The screenshot shows the Burp Suite interface on the left and a browser window on the right. In the Burp Suite 'Proxy' tab, there are several buttons: 'Forward', 'Drop', 'Intercept is on' (which is highlighted in blue), 'Action', and 'Open browser'. The browser window displays a page from 'https://labs.hackify.in/HTML/html\_lab/lab\_6/html\_injection\_6...'. The page has a logo for 'HACKIFY cybersecurity' and the text 'Search and Filter'. A search bar contains the encoded string '%3Ch1%3E+courage%3C%2Fh1%'. Below the search bar, the text 'Your Searched results for ;h1; courage;/h1;' is displayed. At the bottom of the browser window, there is a copyright notice: '© Copyrights 2021 Hackify Cybersecurity All rights reserved'.

The screenshot shows a dual-monitor setup. The left monitor displays the Burp Suite interface, specifically the Proxy tab, with a captured request for 'https://labs.hackify.in:443'. The request details show a POST method to '/HTML/html\_lab/lab\_6/html\_injection\_6.php' with various headers and a large payload containing the string 'search%21%3Ch1%238%2Bcourage%230%252Ph1%253E'. The right monitor displays a browser window for 'https://labs.hackify.in/HTML/html\_lab/lab\_6/html\_injection\_6...'. The page content includes the search term 'h1; courage;/h1;', the Hackify logo with ISO 27001 certification, and a search bar with the same query. Below the search bar, it says 'Your Searched results for ;h1; courage;/h1;'. The bottom right of the browser window has a copyright notice: '© Copyrights 2021 Hackify Cybersecurity All rights reserved'.

# Proof of Concept

The screenshot displays two windows side-by-side. On the left is the Burp Suite interface, specifically the Proxy tab, which is active. The status bar at the bottom of the Burp window indicates "Intercept is on". On the right is a web browser window showing the Hacktify Cybersecurity website at [https://labs.hacktify.in/HTML/html\\_lab/lab\\_6/html\\_injection\\_6...](https://labs.hacktify.in/HTML/html_lab/lab_6/html_injection_6...). The browser's address bar shows the URL. The Hacktify logo is visible at the top of the page. Below it, the main content area features a search bar with the placeholder "Enter text" and a yellow "Search" button. The search results for the term "courage" are displayed, with the word highlighted in black. At the bottom of the page, a copyright notice reads "© Copyrights 2021 Hacktify Cybersecurity All rights reserved".