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.

Application , Name	**{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**
Tools Used	
Burp-Suite is used to find the vulnerability.	
Vulnerability Description	
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.	
How It Was Discovered	
Automated Tools – Browser Inspect Burp-Suite	
Vulnerable URLs	
https://labs.hacktify.in/HTML/open_redirect_lab/lab_1/index.php	
Consequences of not Fixing the Issue	
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).	
Suggested Countermeasures	

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.

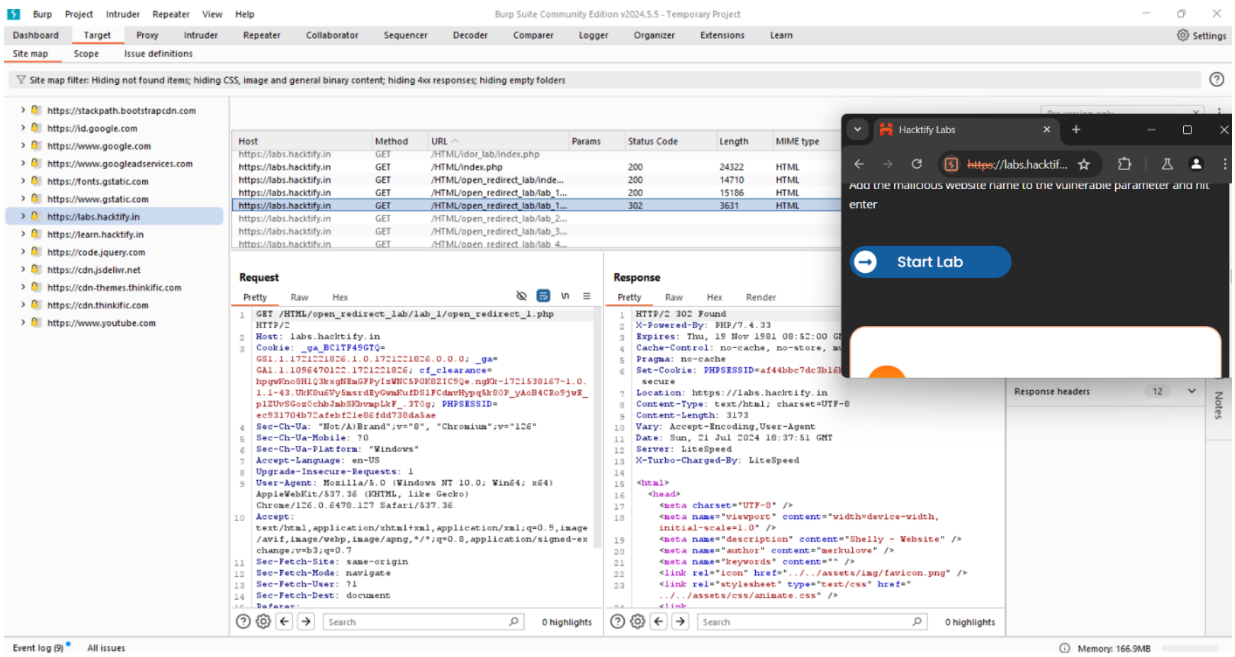
Frame Busting Scripts: Implement JavaScript frame-busting techniques to prevent your site from being loaded in an iframe.

References

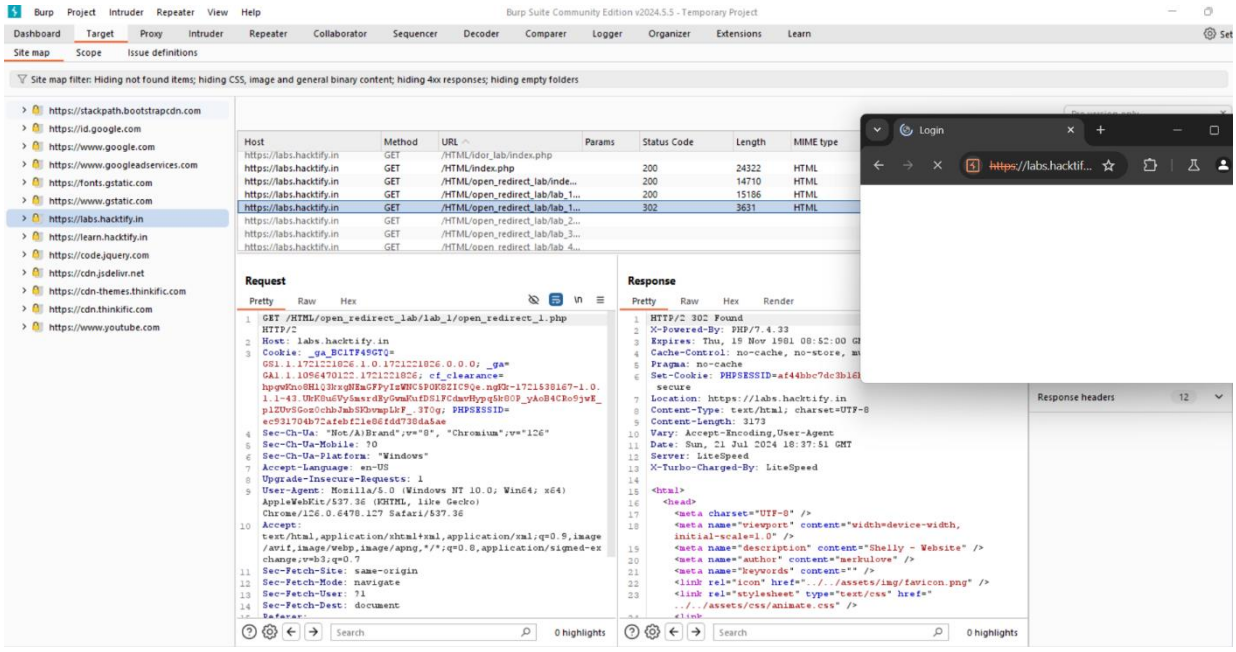
https://www.coursera.org/
https://portswigger.net/web-security/clickjacking
https://owasp.org/www-community/attacks/Clickjacking
https://www.imperva.com/learn/application-security/clickjacking/

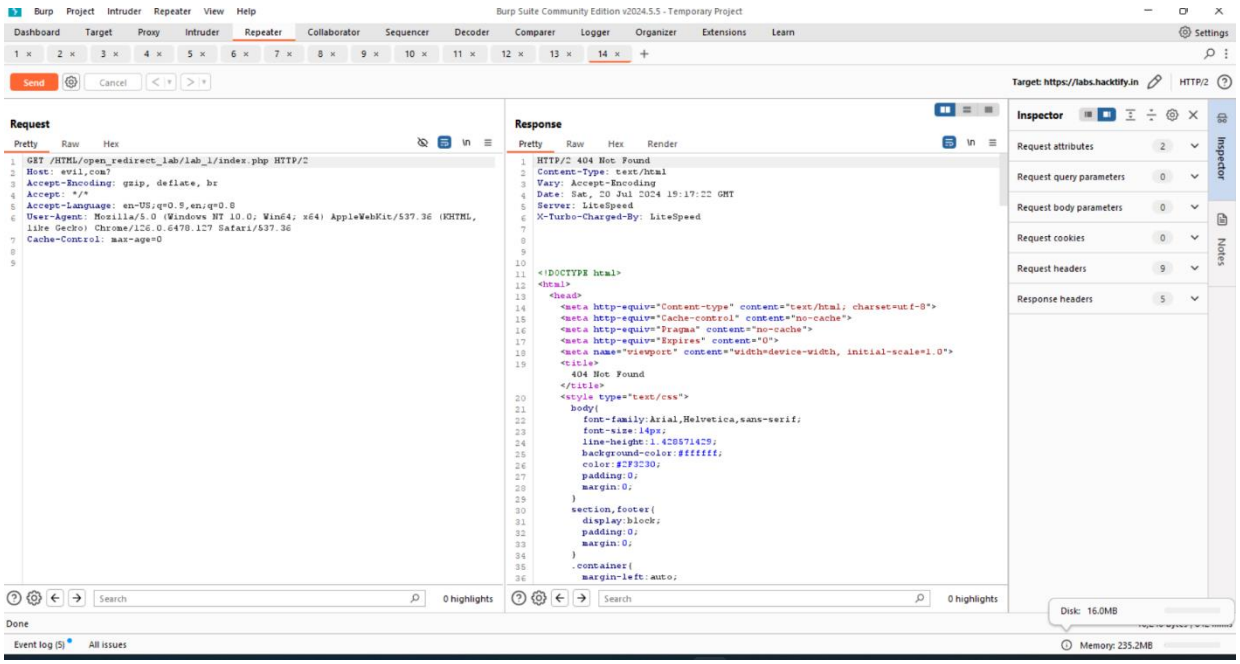
This section contains the proof of the above vulnerabilities as the screenshot of the vulnerability of the lab

Before starting the lab.



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



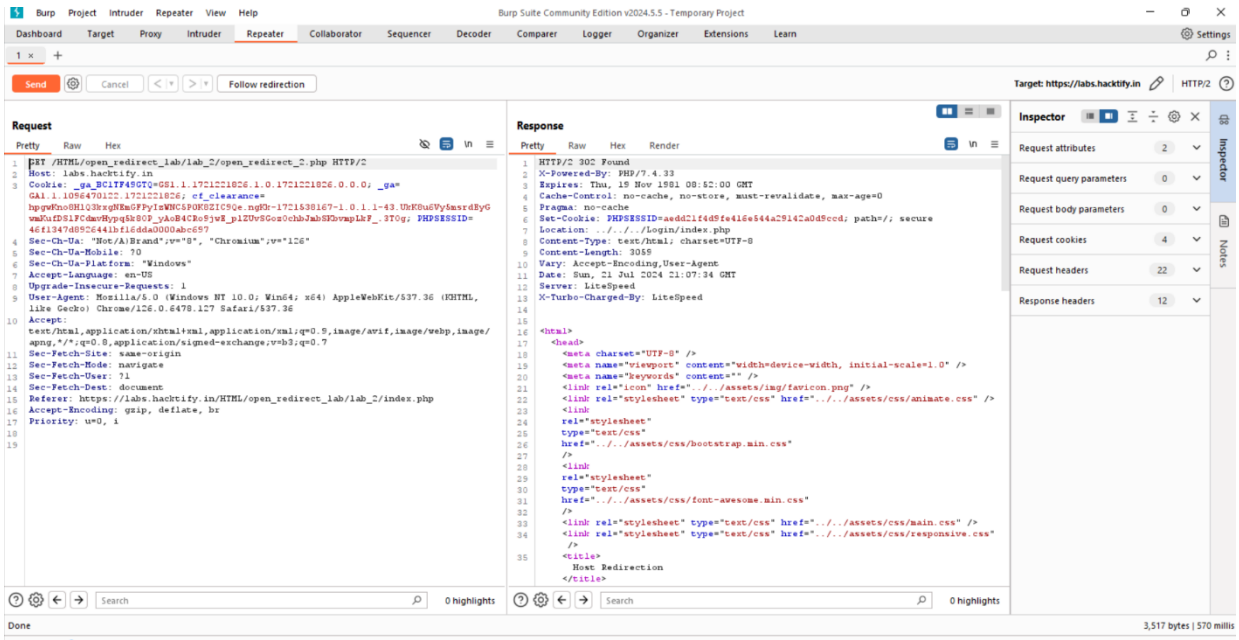


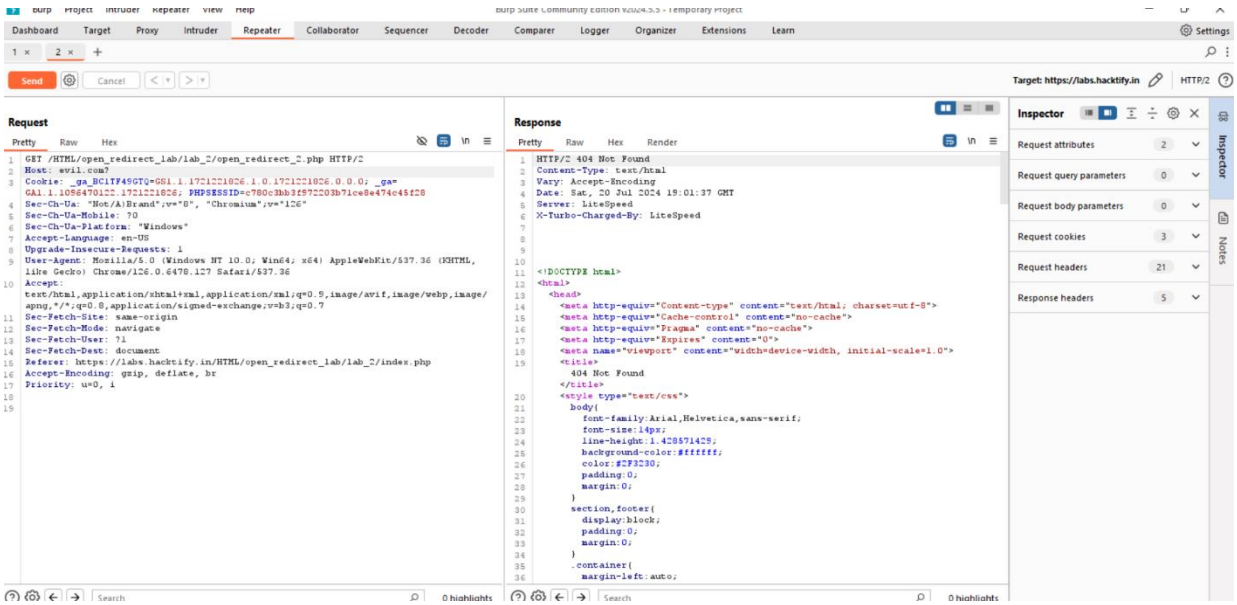
Proof of Concept

1.2. Story Of a Beautiful Header!

Reference	Risk Rating
Sub-lab-2 Story Of a Beautiful Header!	**Low**
Tools Used	
Burp-Suite, Browser.	
Vulnerability Description	
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.	
How It Was Discovered	
Automated Tools – Browser Inspect	
Vulnerable URLs	
https://labs.hacktify.in/HTML/open_redirect_lab/lab_2/index.php	
Consequences of not Fixing the Issue	
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.	
Suggested Countermeasures	
Prompt users for confirmation before redirecting them to an external site. This adds an extra layer of security by ensuring users are aware of the redirection.	
References	
https://portswigger.net/web-security/clickjacking https://owasp.org/www- security/clickjacking/	

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





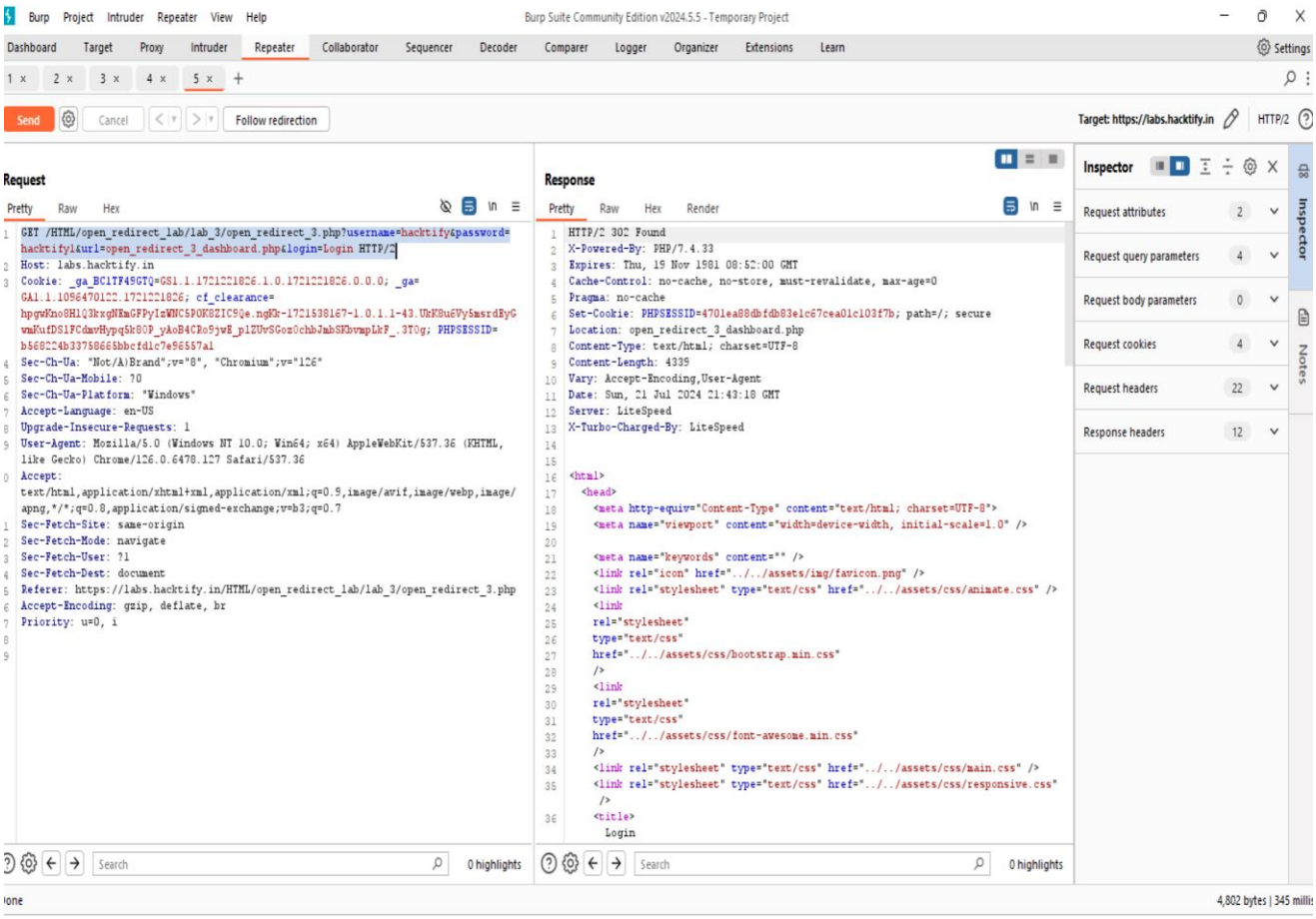
Proof of Concept

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	
https://labs.hacktify.in/HTML/open_redirect_lab/lab_3/open_redirect_3.php	
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	
1. **Use a Whitelist**: Implement a whitelist of allowed URLs for redirection. This ensures that only predefined, safe URLs can be used. 2. **Validate and Sanitize Input**: Ensure that any input used in redirects is validated and sanitized. Only allow redirects to trusted URLs.	
References	
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

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

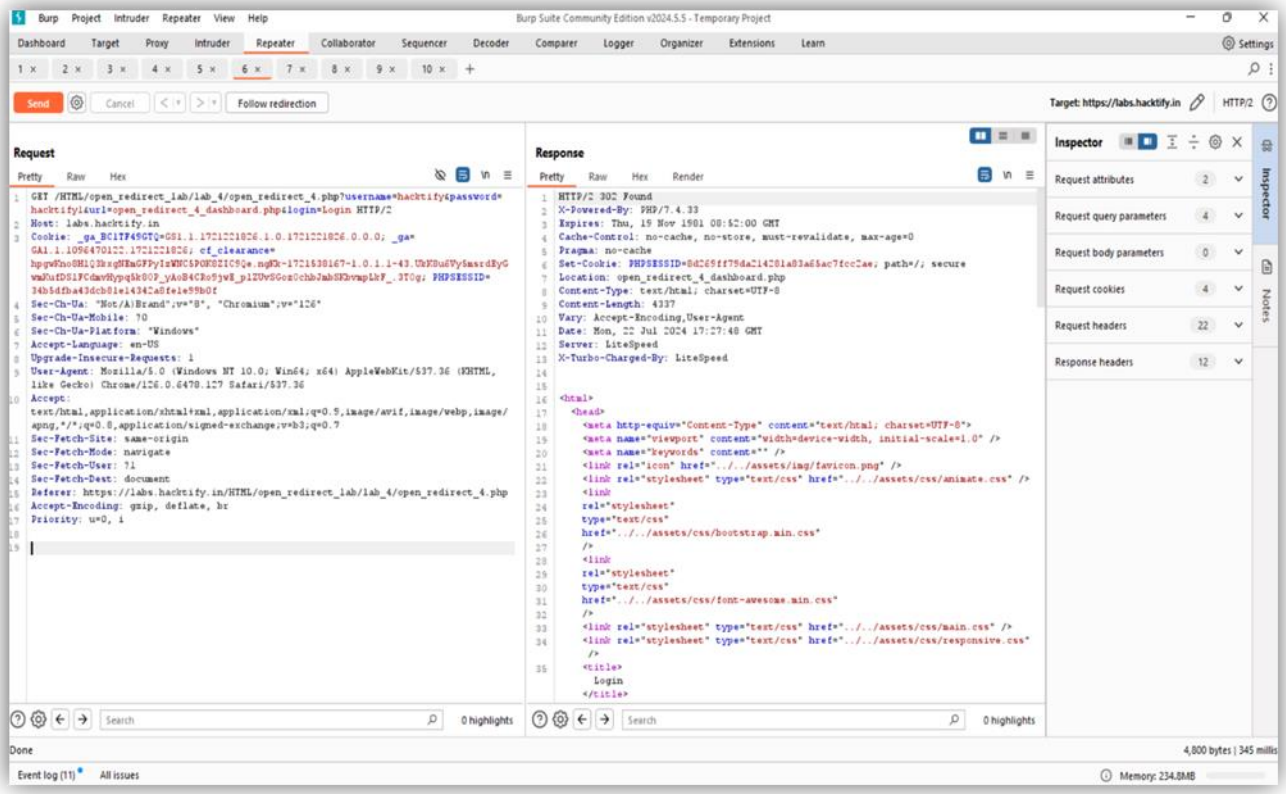


Proof of Concept

1.4. Patterns are Important!

Reference	Risk Rating
Patterns are Important	Medium
Tools Used	
Burp-Suite is used to find the vulnerability.	
Vulnerability Description	
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	
How It Was Discovered	
Automated Tools –Burp-suite	
Vulnerable URLs	
https://labs.hacktify.in/HTML/open_redirect_lab/lab_4/open_redirect_4.php	
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	
https://portswigger.net/web-security/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



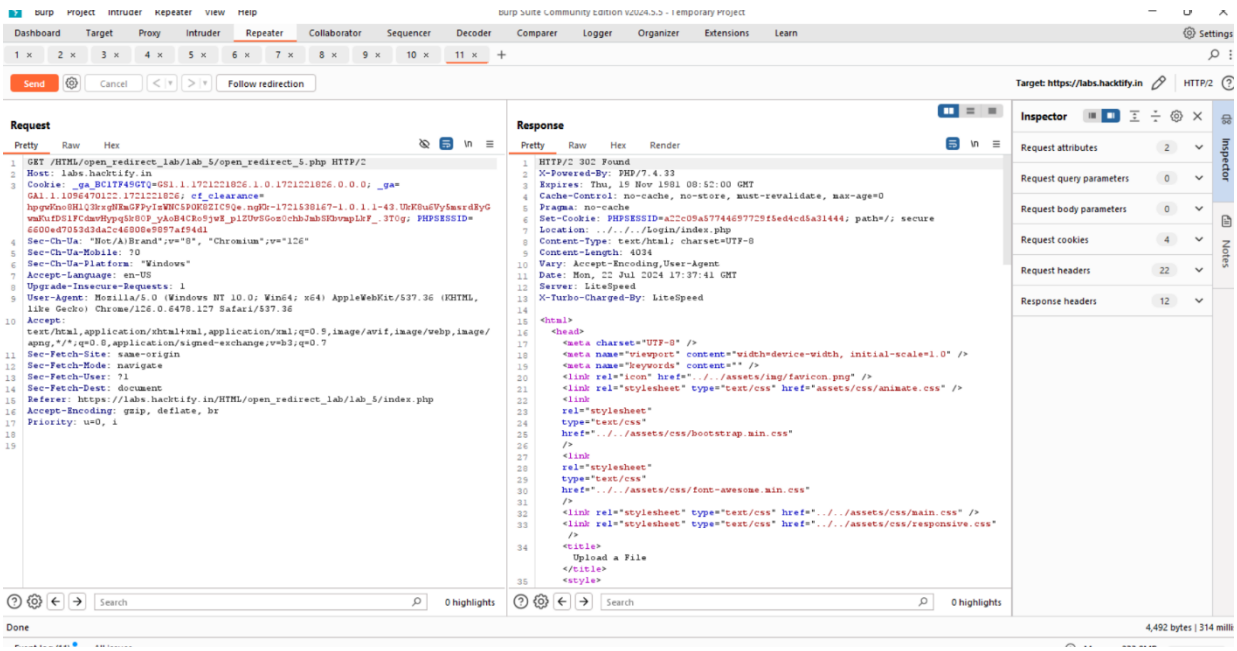
1.5. File Upload!?Redirect IT!

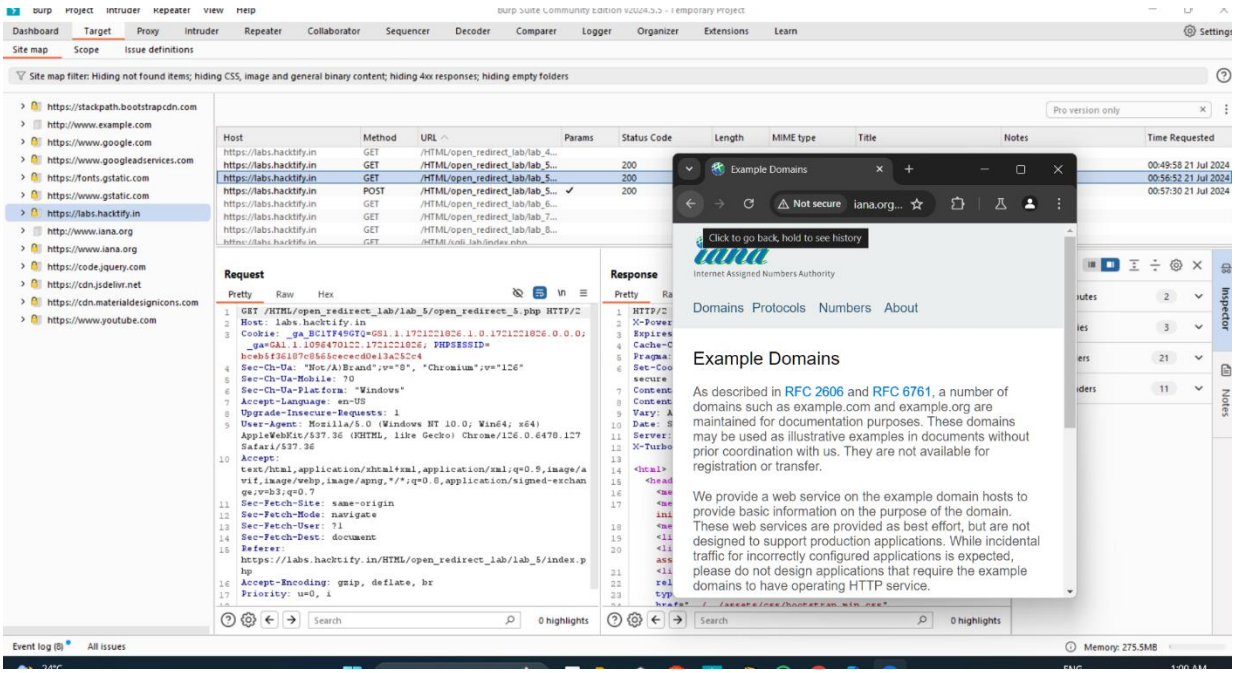
Reference	Risk Rating
Sub-lab-1: File Uplaod!?Redirect IT!	**Low**
Tools Used	
Burp-Suite is used to find the vulnerability.	
Vulnerability Description	
Accepts any kind of extension file.	
How It Was Discovered	

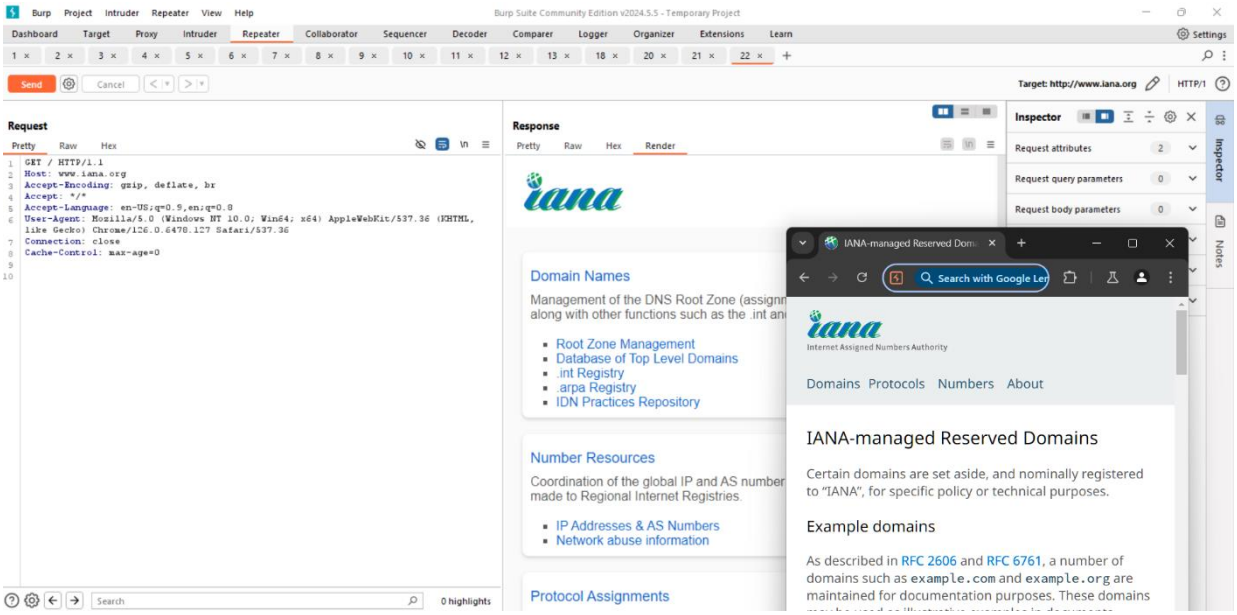
Proof of Concept

Automated Tools – Browser Inspect Burp suite
Vulnerable URLs
https://labs.hacktify.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







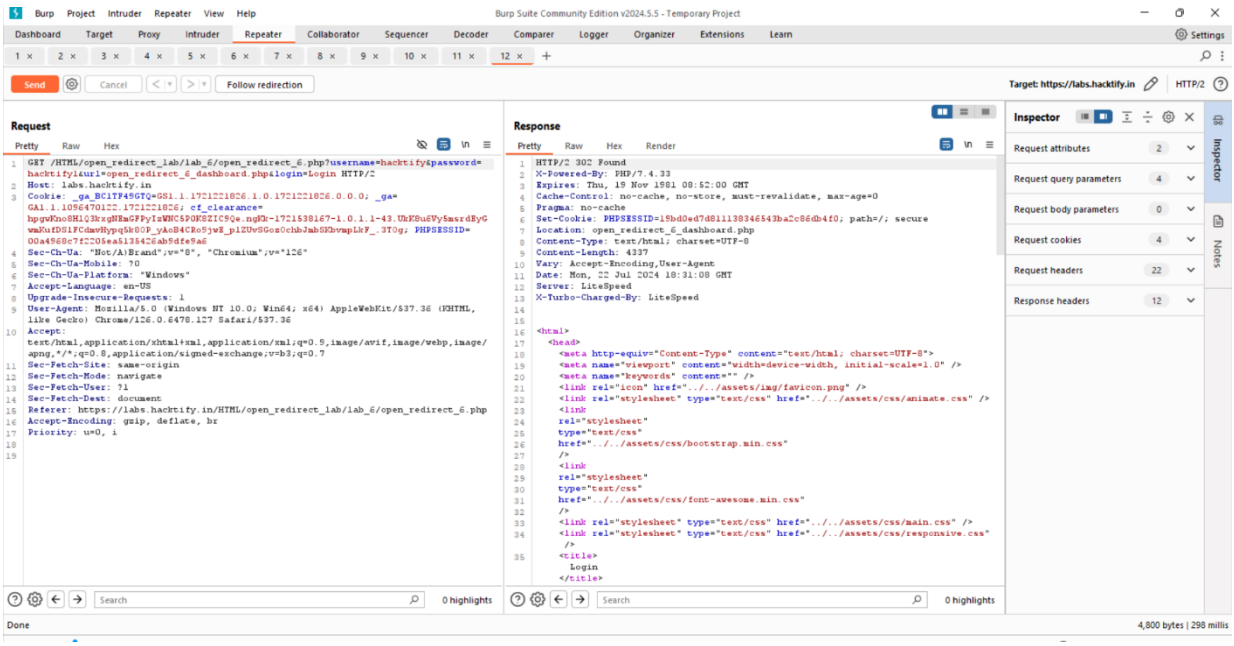
1.6. Same Param Twice!

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 **open redirect**, 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
https://labs.hacktify.in/HTML/open_redirect_lab/lab_6/open_redirect_6.php
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
https://portswigger.net/web-security/clickjacking

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

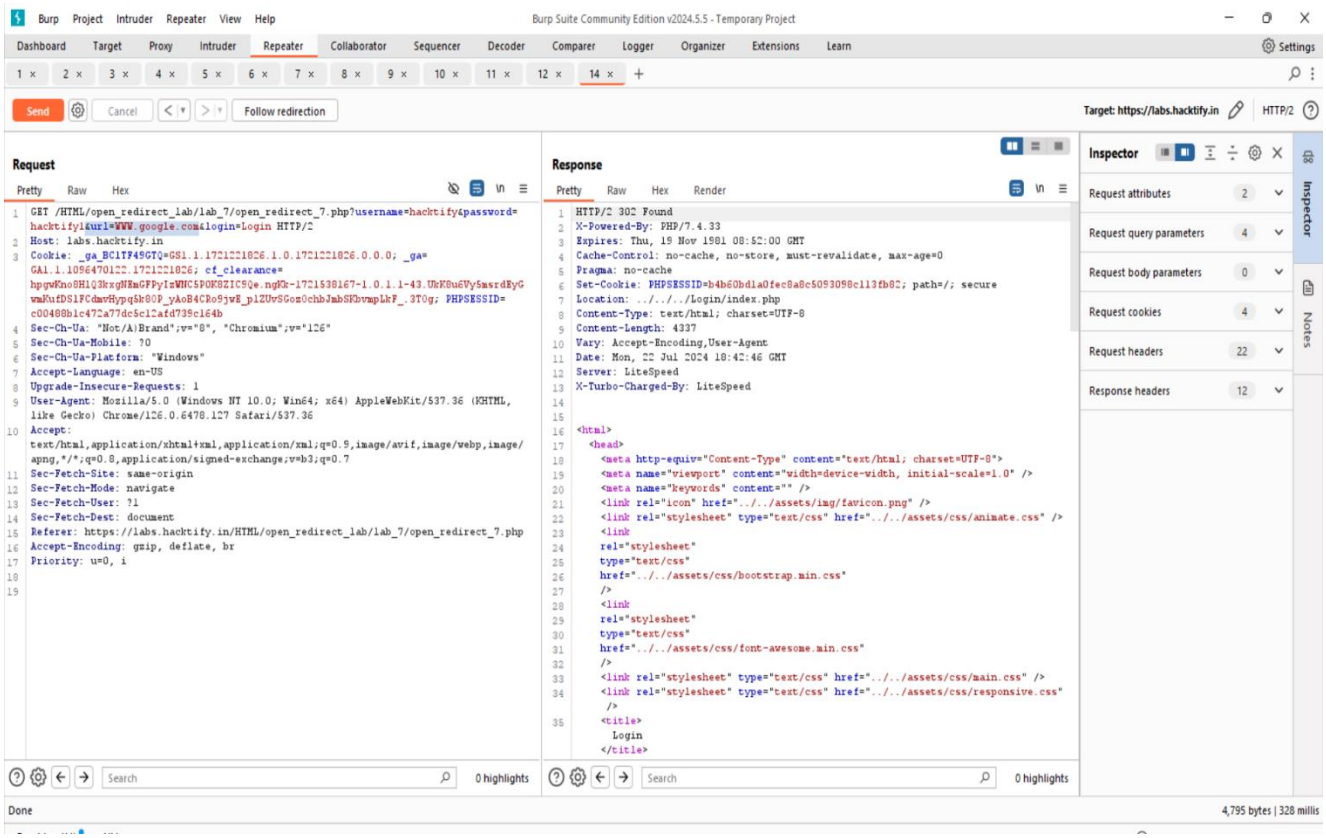


1.7. Domains? Not Always!

Reference	Risk Rating
Sub-lab-7: Domain? Not Always	**High**
Tools Used	
Burp-Suite	
Vulnerability Description	
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.	
How It Was Discovered	
Automated Tools –Burp Suite (Browser inspect)	
Vulnerable URLs	
https://labs.hacktify.in/HTML/open_redirect_lab/lab_7/open_redirect_7.php	
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	
Avoid allowing user input to determine the destination URL. Instead, use server-side logic to handle redirections.	
References	
https://portswigger.net/web-security/clickjacking https://brightsec.com/blog/open-redirect-vulnerabilities/	

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

Proof of Concept



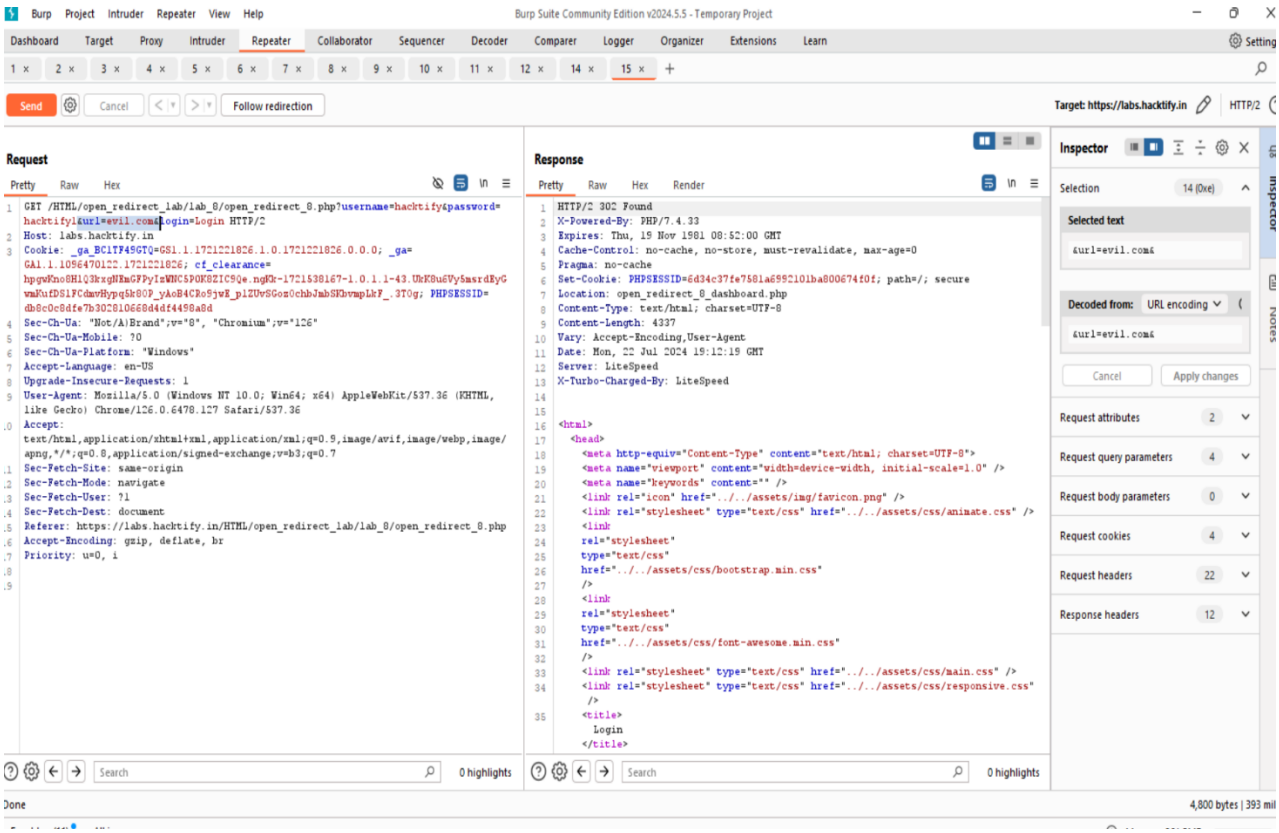
1.8. Style Digit Symbols<3

Reference	Risk Rating

Sub-lab-8: Style Digit Symbols	High
Tools Used	
Burp-Suite	
Vulnerability Description	
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.	
How It Was Discovered	
Automated Tools –Burp Suite (Browser inspect)	
Vulnerable URLs	
https://labs.hacktify.in/HTML/open_redirect_lab/lab_8/open_redirect_8.php	
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	
Where possible, use relative URLs instead of absolute URLs to prevent redirection to external sites. **Use security headers like Content-Security-Policy (CSP) to restrict the sources that can be loaded. This helps prevent unauthorized redirections**	
References	
https://portswigger.net/web-security/clickjacking https://dzone.com/articles/what-is-an-open-redirection-vulnerability-and-how	

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

Proof of Concept



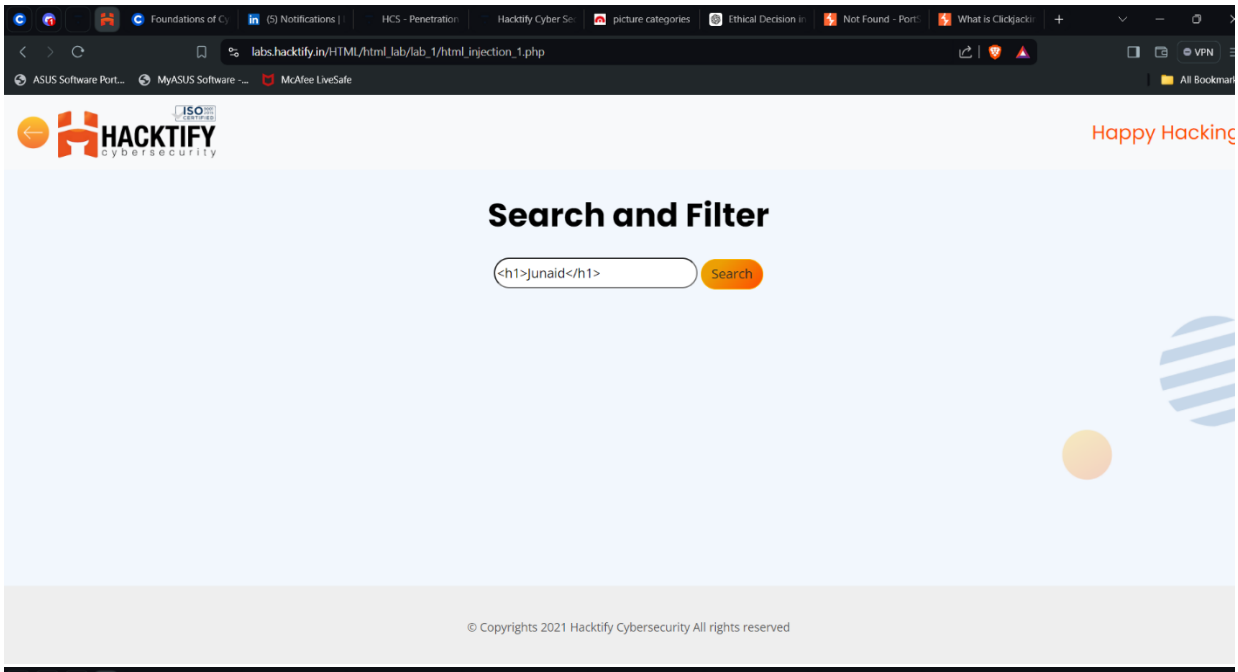
3. HTML Injection

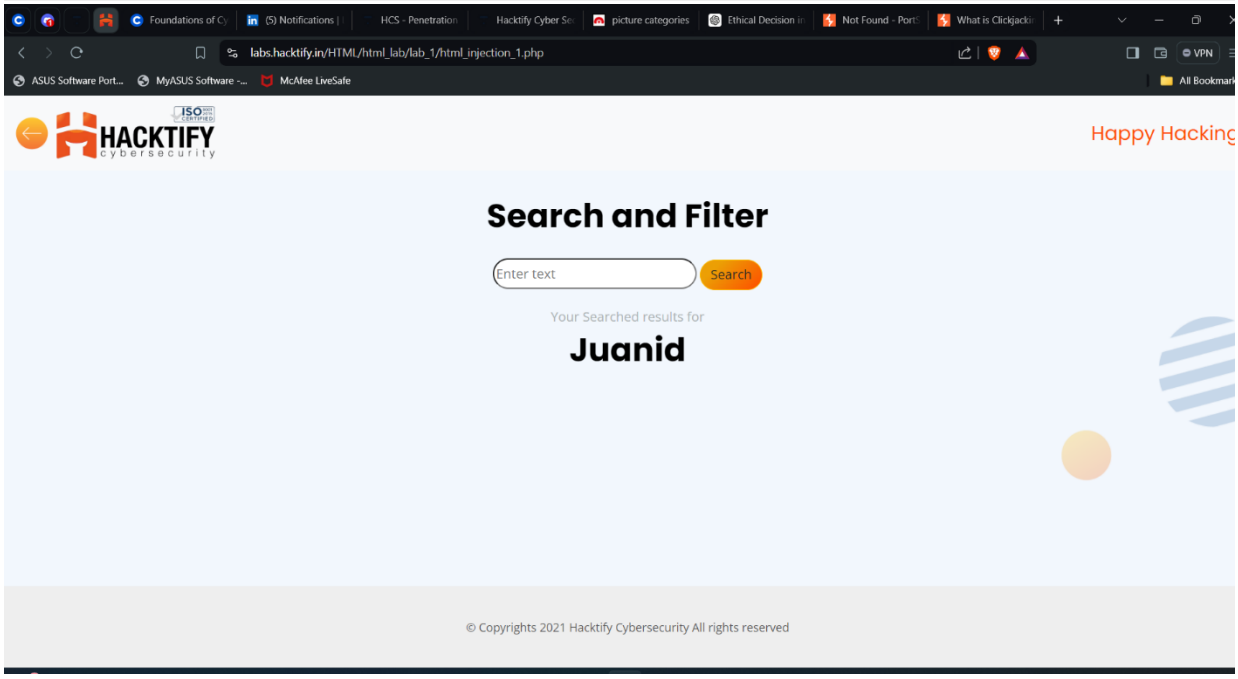
2.1. HTML's are easy!

Reference	Risk Rating
Sub-lab-1: HTML's are easy!	**Low**
Tools Used	
Browser *"View Page Sources"* is used to find the vulnerability.(Also inspect)	
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)	
Vulnerable URLs	
https://labs.hacktify.in/HTML/html_lab/lab_1/index.php	
Consequences of not Fixing the Issue	
Malicious HTML can be used to steal session cookies, allowing attackers to impersonate users and gain unauthorized access to their accounts.	
Suggested Countermeasures	
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.	
References	
https://trainings.internshala.com/blog/html-injections/ https://owasp.org/www-community/Injection_Information https://en.wikipedia.org/wiki/HTML_injection	

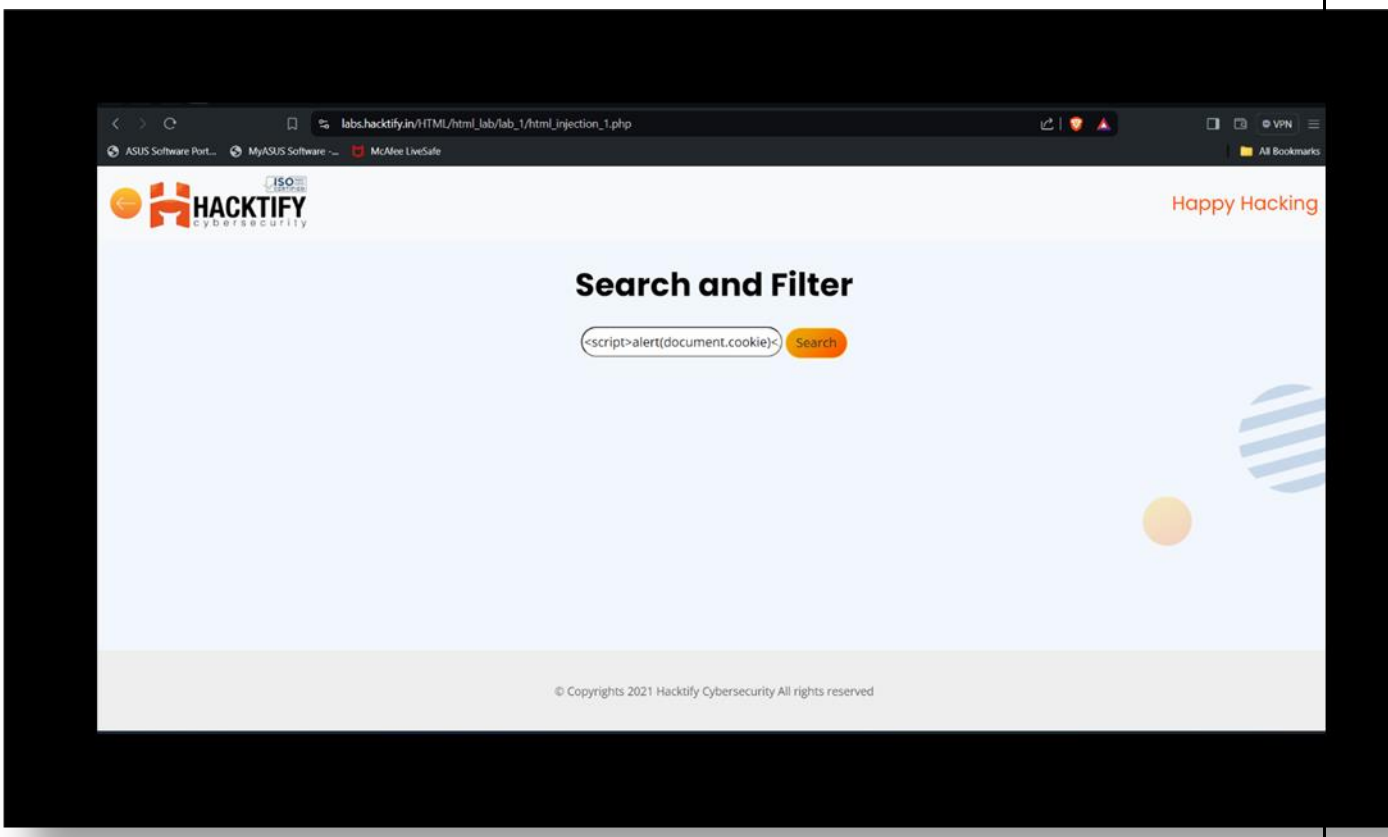
This section contains the proof of the above vulnerabilities as the screenshot of the vulnerability of the lab

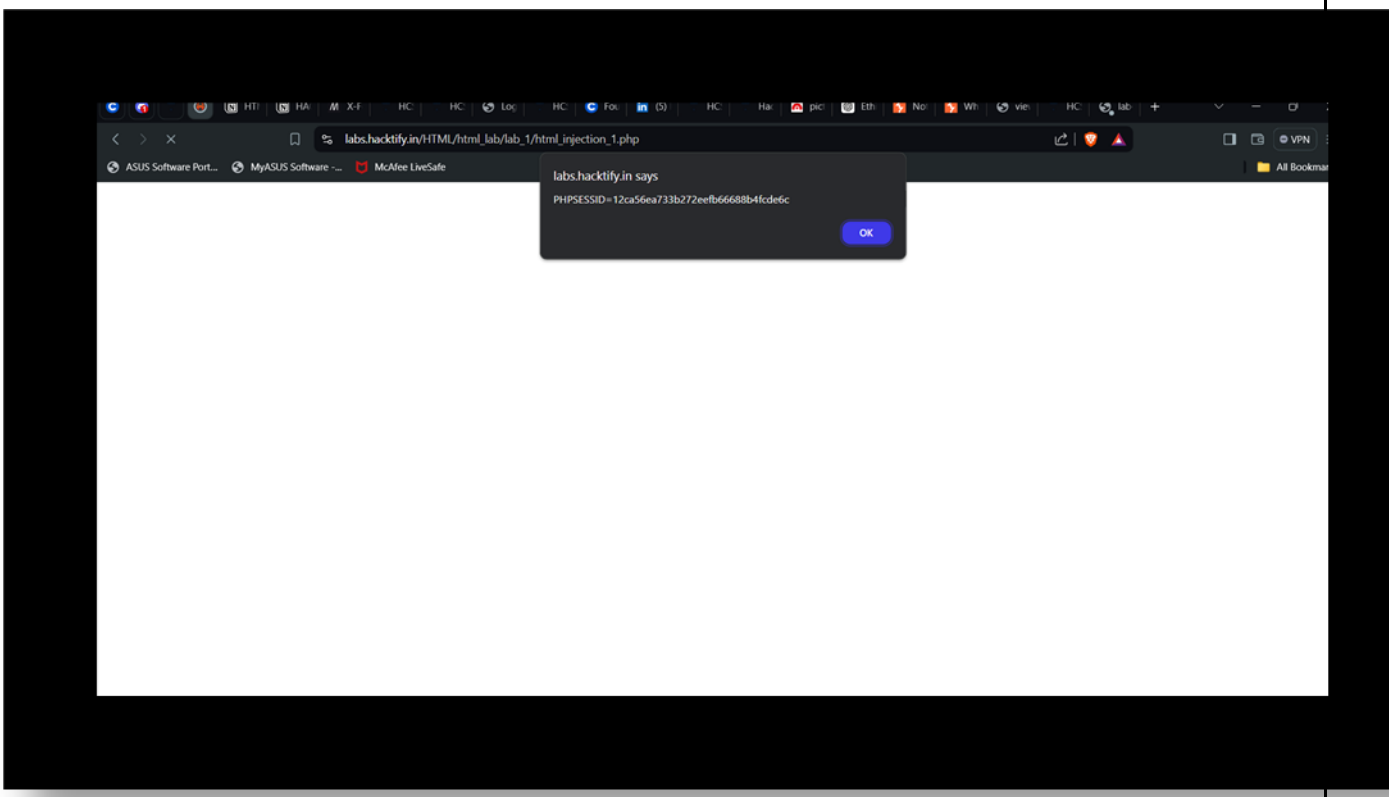
Proof of Concept





=>


labs.hacktify.in/HTML/html_lab/lab_1/html_injection_1.php
ASUS Software Port...
MyASUS Software -...
McAfee LiveSafe
All Bookmarks
HACKTIFY
cybersecurity
Happy Hacking
Search and Filter
<script>alert(document.cookie)<
Search
© Copyrights 2021 Hacktify Cybersecurity All rights reserved

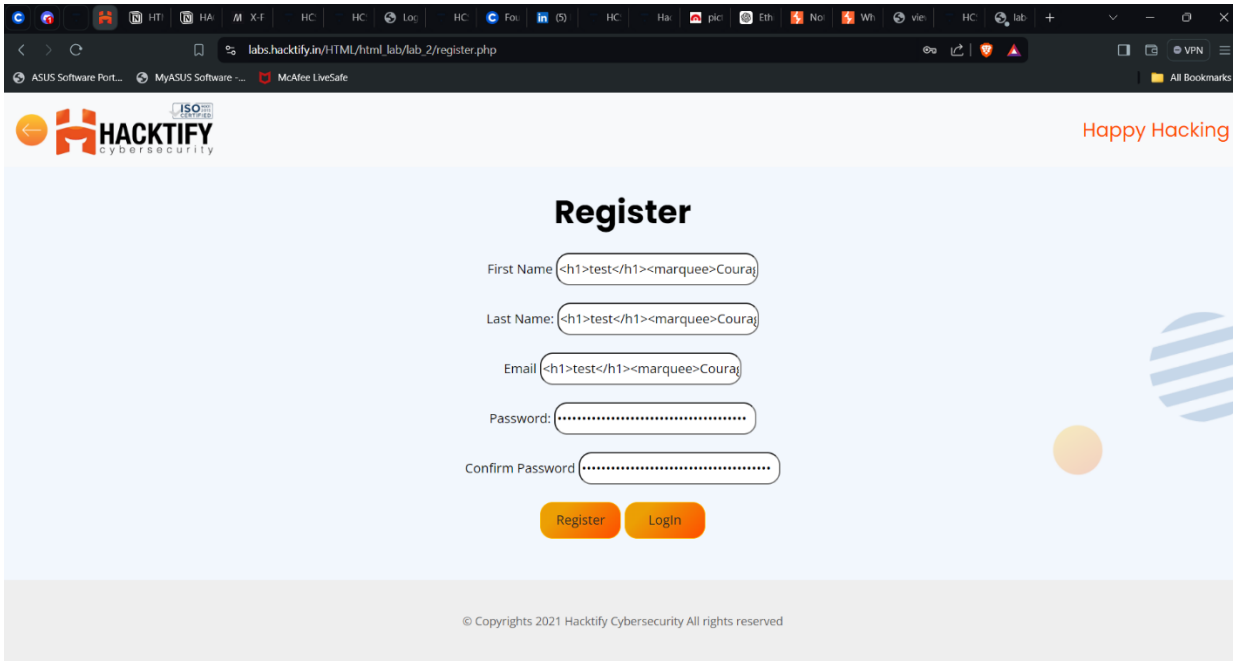

labs.hacktify.in/HTML/html_lab/lab_1/html_injection_1.php
ASUS Software Port...
MyASUS Software -...
McAfee LiveSafe
labs.hacktify.in says
PHPSESSID=12ca56ea733b272eefb66688b4fcde6c
OK

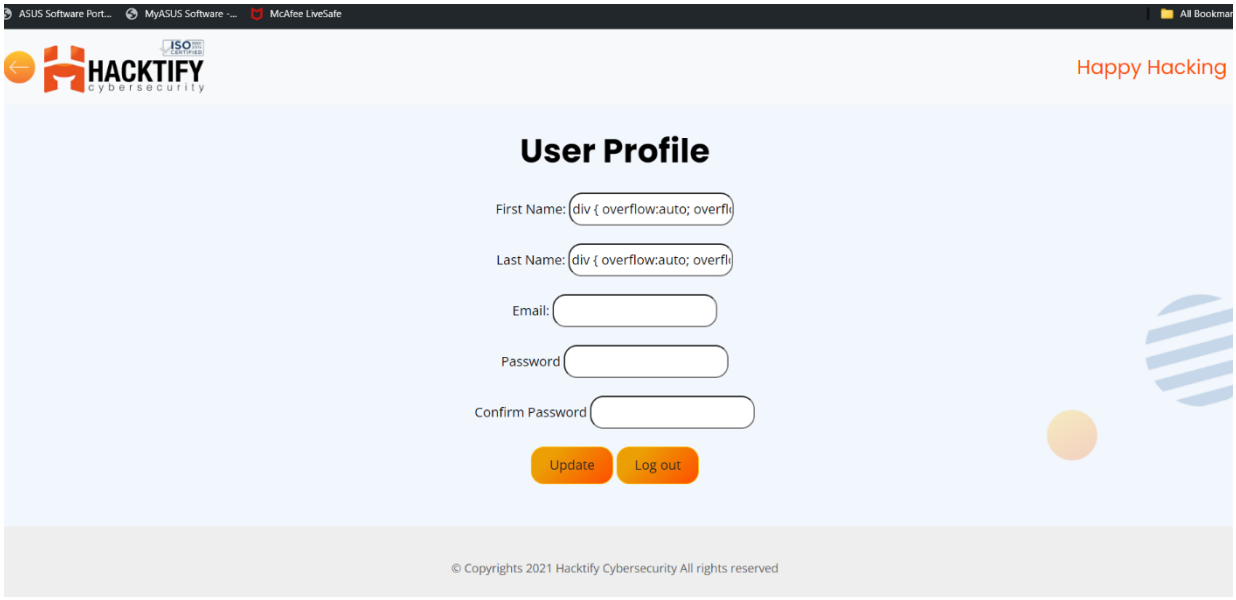
Proof of Concept

2.2. Let me Store them!

Reference	Risk Rating
Sub-lab-1: Let me Store them	**Low**
Tools Used	
Browser "*View Page Sources*" is used to find the vulnerability.	
Vulnerability Description	
Automated Tools – Browser View Page Sources (Ctrl + U)	
Vulnerable URLs	
https://labs.hacktify.in/HTML/html_lab/lab_2/index.php	
Consequences of not Fixing the Issue	
Attackers can alter the appearance of your website, damaging your brand's reputation and causing users to lose trust in your site.	
Suggested Countermeasures	
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	
References	
https://owasp.org/www-community/Injection_Information https://en.wikipedia.org/wiki/HTML_injection	

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


labs.hacktify.in/HTML/html_lab/lab_2/register.php
ASUS Software Port...
MyASUS Software -...
McAfee LiveSafe
HACKTIFY
cybersecurity
Happy Hacking
Register
First Name
<h1>test</h1><marquee>Coura
Last Name:
<h1>test</h1><marquee>Coura
Email
<h1>test</h1><marquee>Coura
Password:
Confirm Password
Register
Login
© Copyrights 2021 Hacktify Cybersecurity All rights reserved

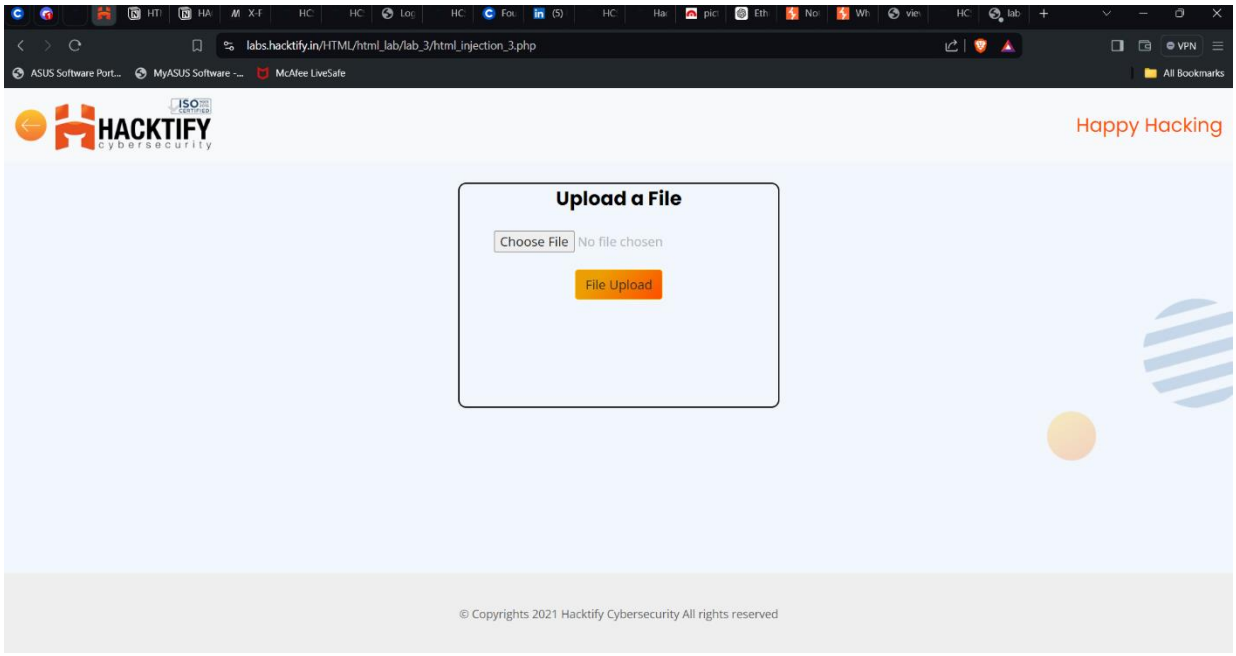

ASUS Software Port...
MyASUS Software -...
McAfee LiveSafe
HACKTIFY
cybersecurity
Happy Hacking
User Profile
First Name:
div { overflow:auto; overfl
Last Name:
div { overflow:auto; overfl
Email:
Password
Confirm Password
Update
Log out
© Copyrights 2021 Hacktify Cybersecurity All rights reserved

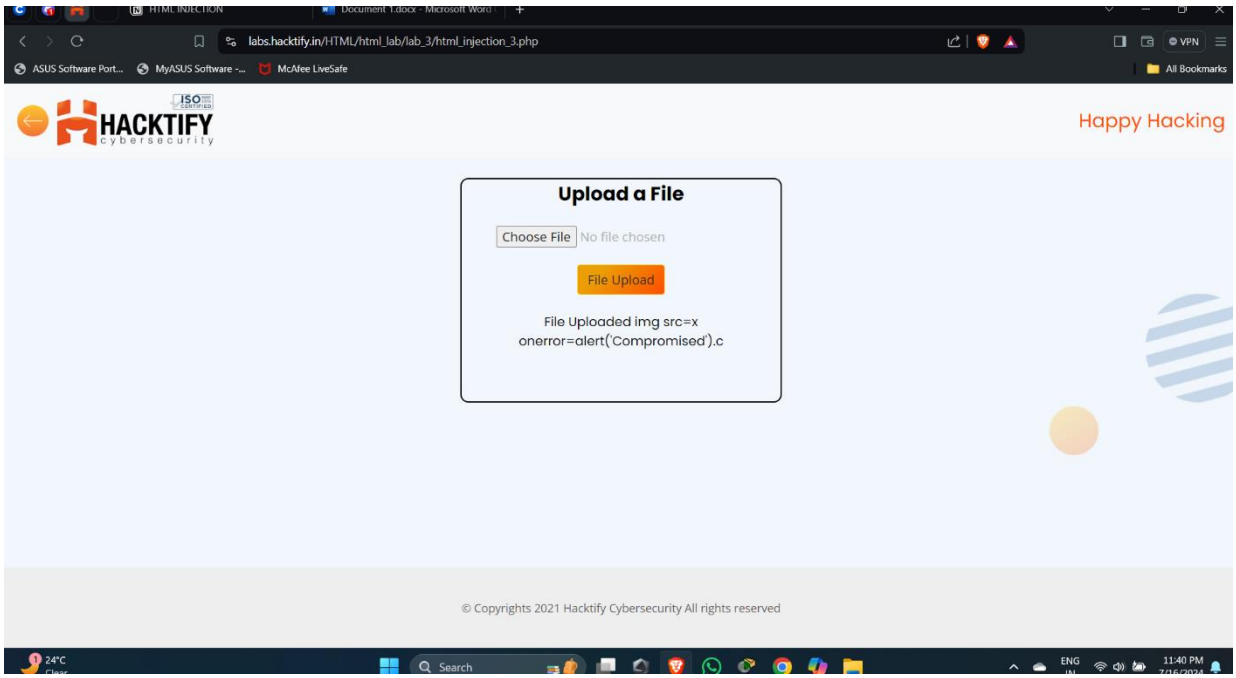
Proof of Concept

2.3. Files Names are also vulnerable!

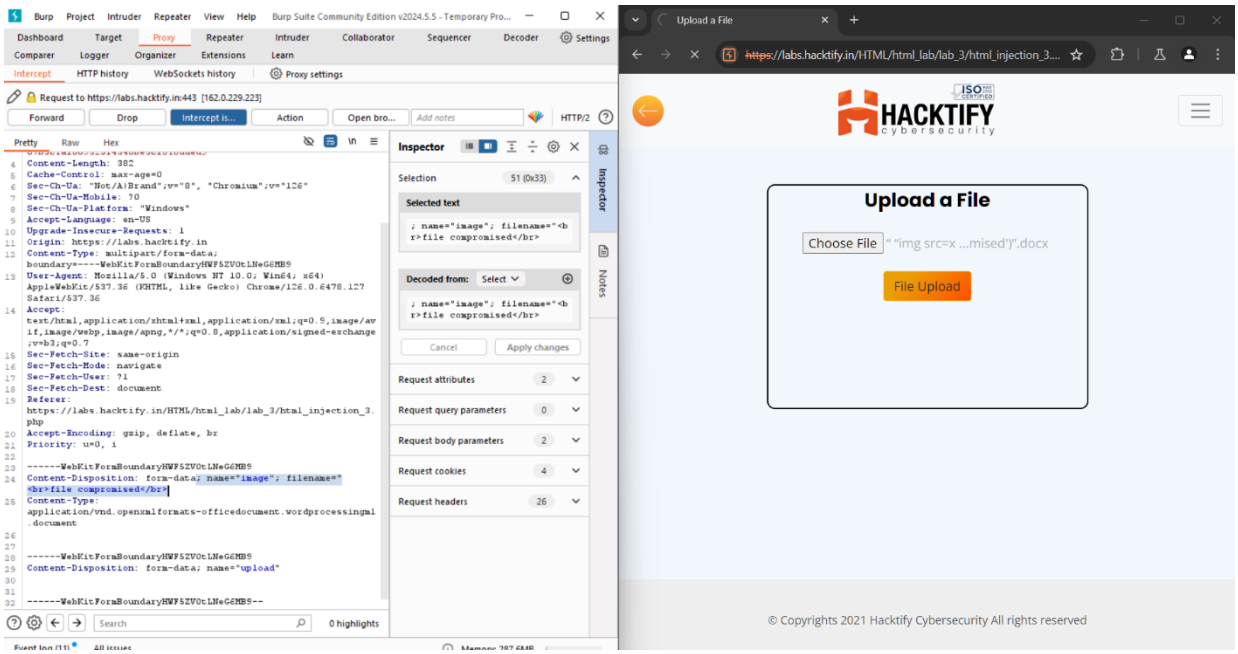
Reference	Risk Rating
Sub-lab-1: Files Names are also vulnerable	**Low**
Tools Used	
Burp-Suite	
Vulnerability Description	
Allowing attackers to upload files containing malicious HTML code. This can lead to various security issues, including data breaches, malware distribution, and server compromise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

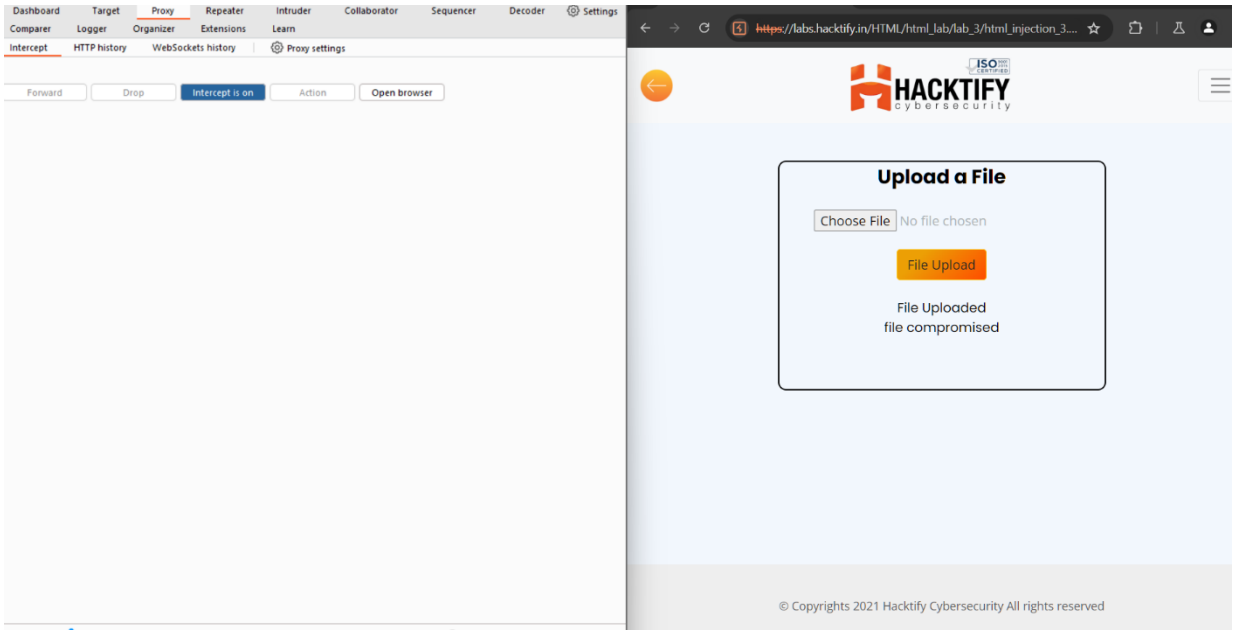
This section contains the proof of the above vulnerabilities as the screenshot of the vulnerability of the lab


labs.hacktify.in/HTML/html_lab/lab_3/html_injection_3.php
ASUS Software Port...
MyASUS Software -...
McAfee LiveSafe
All Bookmarks
HACKTIFY
cybersecurity
Happy Hacking
Upload a File
Choose File
No file chosen
File Upload
© Copyrights 2021 Hacktify Cybersecurity All rights reserved


labs.hacktify.in/HTML/html_lab/lab_3/html_injection_3.php
ASUS Software Port...
MyASUS Software -...
McAfee LiveSafe
All Bookmarks
HACKTIFY
cybersecurity
Happy Hacking
Upload a File
Choose File
No file chosen
File Upload
File Uploaded img src=x
onerror=alert('Compromised').c
© Copyrights 2021 Hacktify Cybersecurity All rights reserved
Search

Proof of Concept



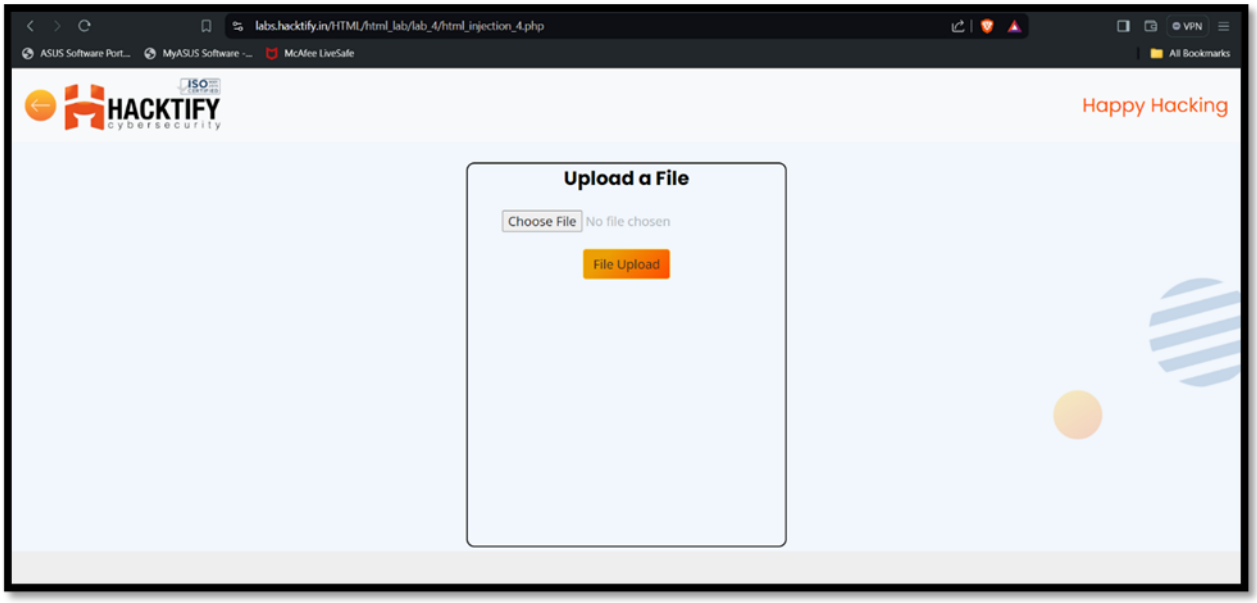


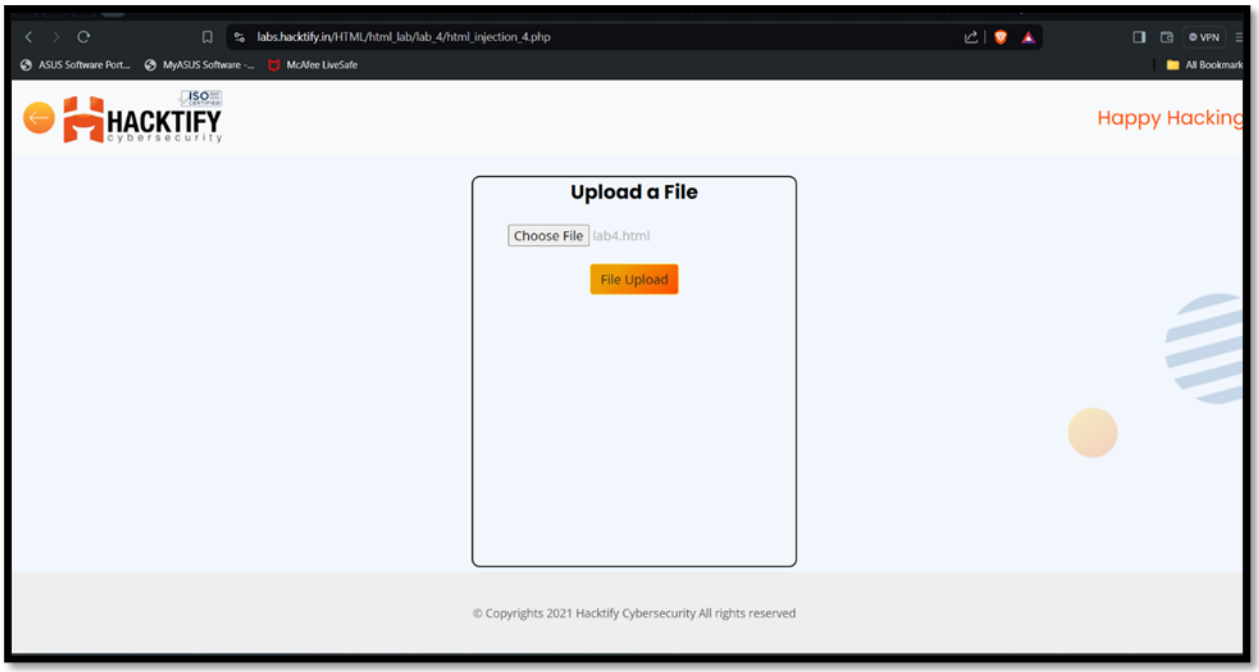
2.4. File Content and HTML Injection a perfect pair!

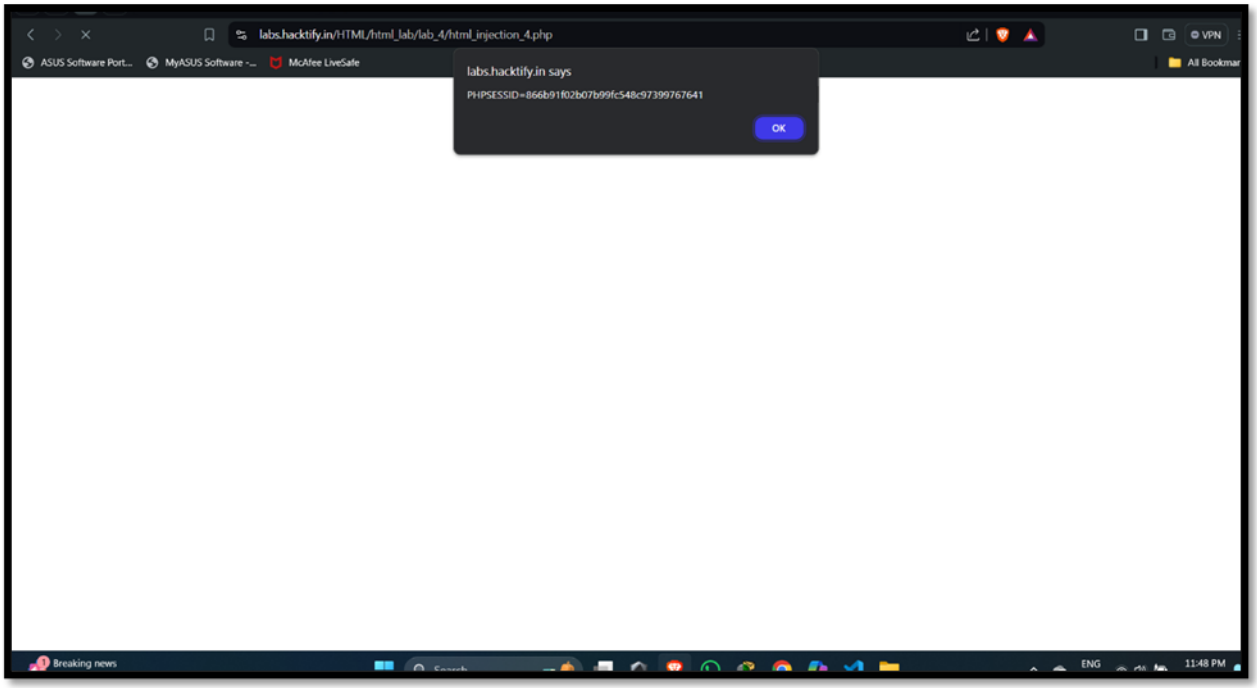
Reference	Risk Rating
Sub-lab-4: File Content and HTML Injection a perfect pair	Medium
Tools Used	
Browser "*View Page Sources*" is used to find the vulnerability.	
Vulnerability Description	
The application fails to properly validate the file type, size, or content.	
How It Was Discovered	
Automated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

Proof of Concept





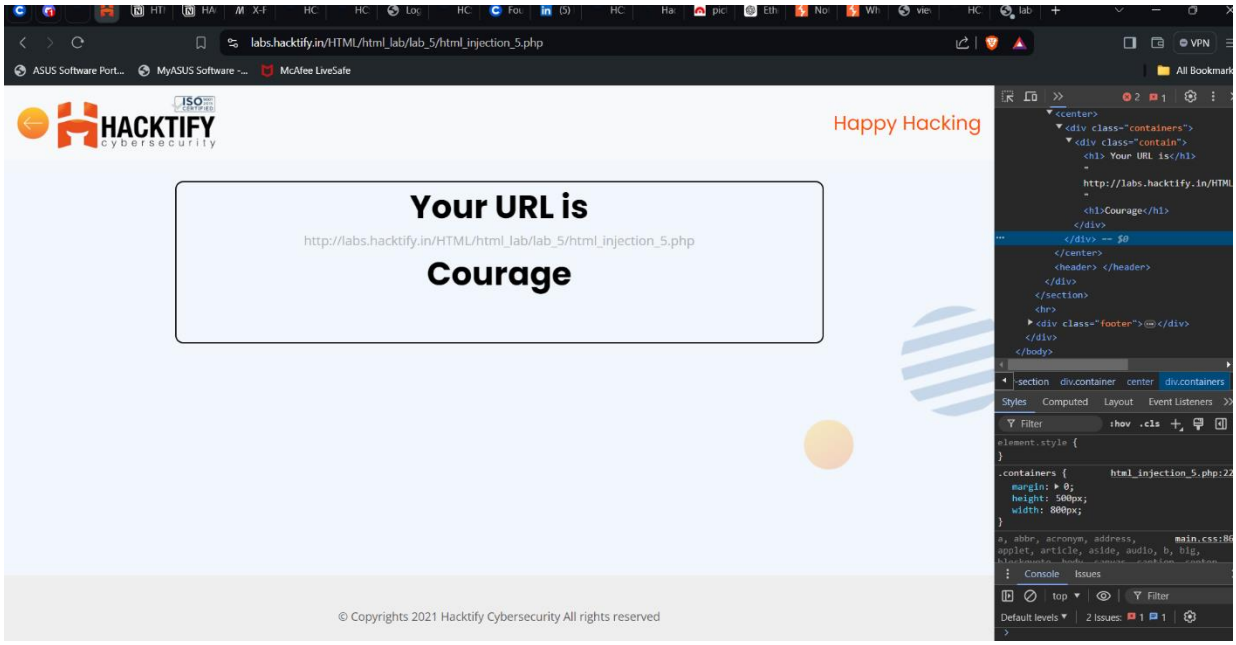


2.5. Injecting HTML using URL

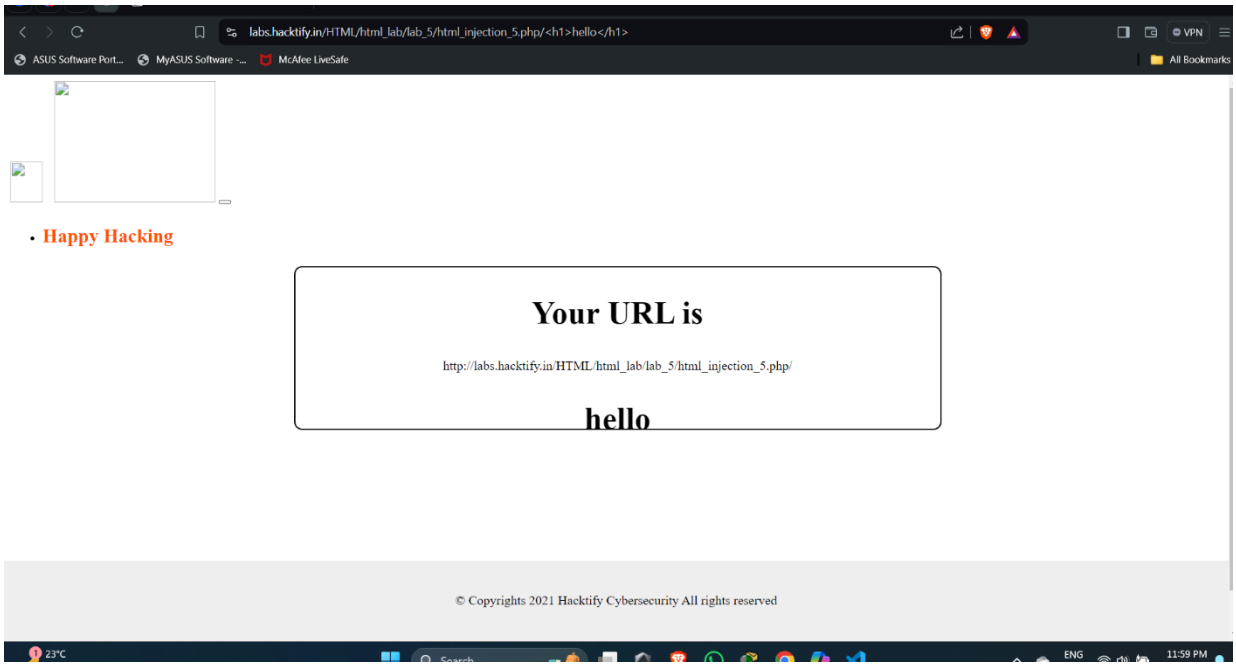
Reference	Risk Rating
Sub-lab-5: Injecting HTML using URL	**Medium**
Tools Used	
Browser "*View Page Sources*" is used .	
Vulnerability Description	
The attacker finds an input point in the web application where they can insert HTML code.	
How It Was Discovered	
Automated Tools – Browser View Page Sources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

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://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
Tools Used	
Burp Suite	
Vulnerability Description	
The attacker finds an input point in the web application where they can insert HTML code. And he can also insert encoded html tags	
How It Was Discovered	
Automated Tool----Burp-suite	
Vulnerable URLs	
https://labs.hacktify.in/HTML/html_lab/lab_6/index.php	
Consequences of not Fixing the Issue	
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.	
Suggested Countermeasures	

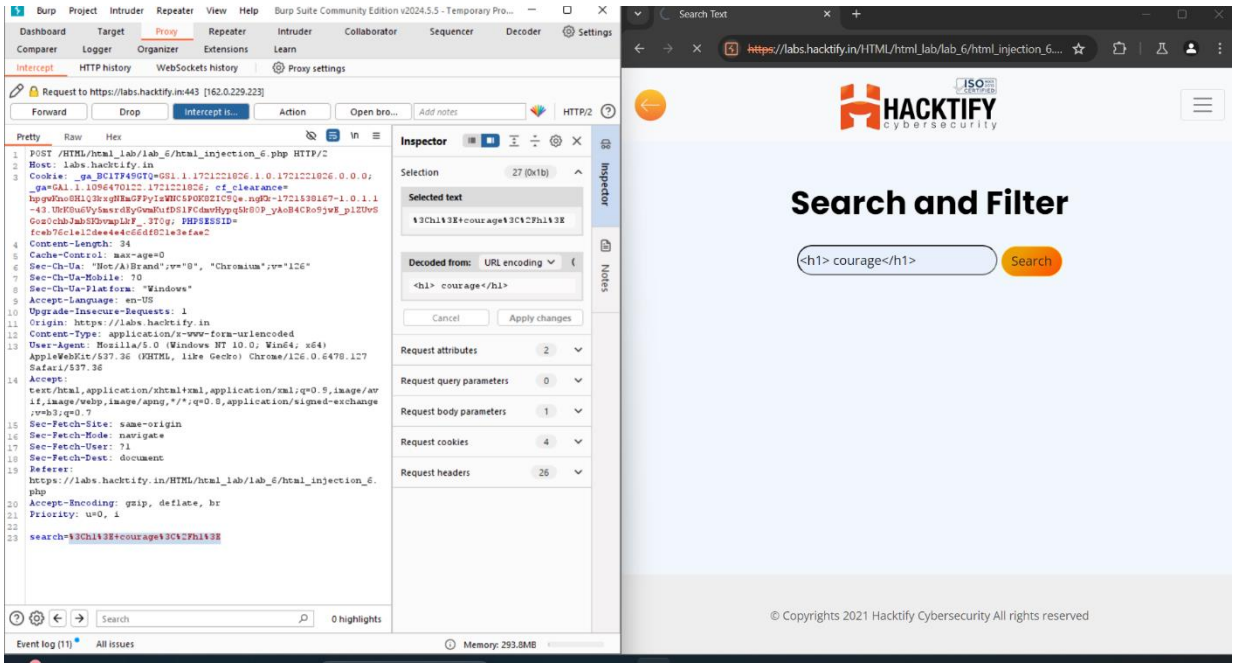
Proof of Concept

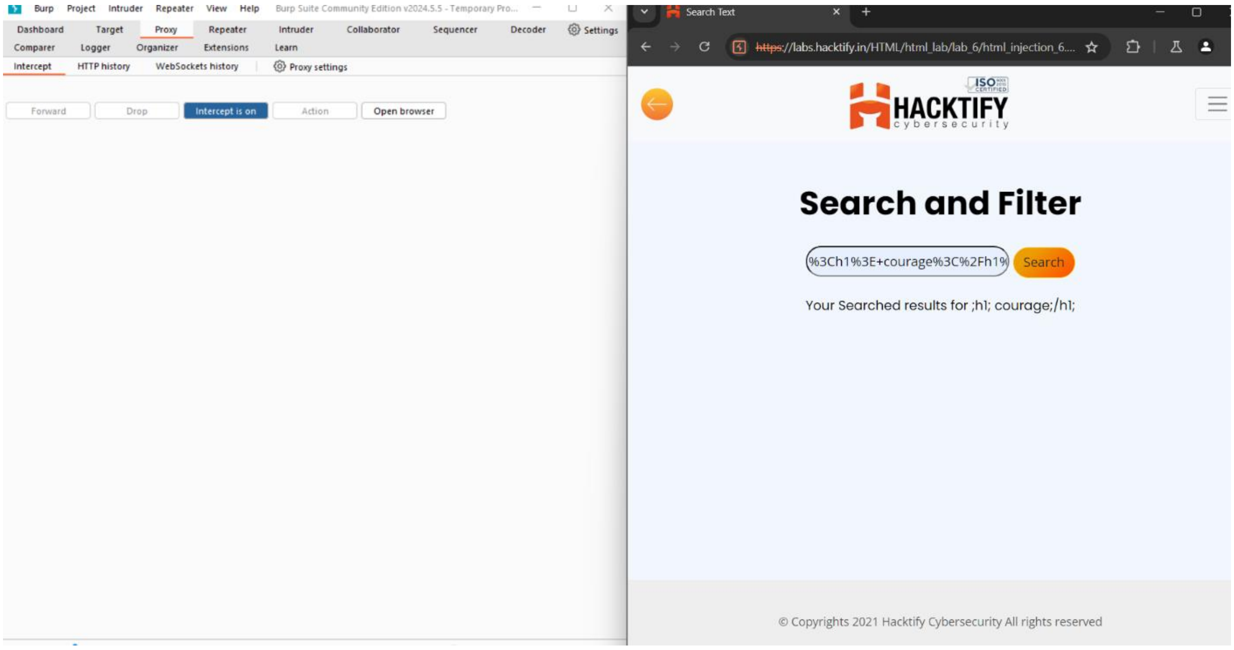
1. **Input Validation and Sanitization**: Rigorously validate and sanitize all user inputs to ensure they adhere to expected formats. This helps prevent malicious HTML from being processed.
2. **Content Security Policy (CSP)**: Implement a strong CSP to control the sources from which content can be loaded.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

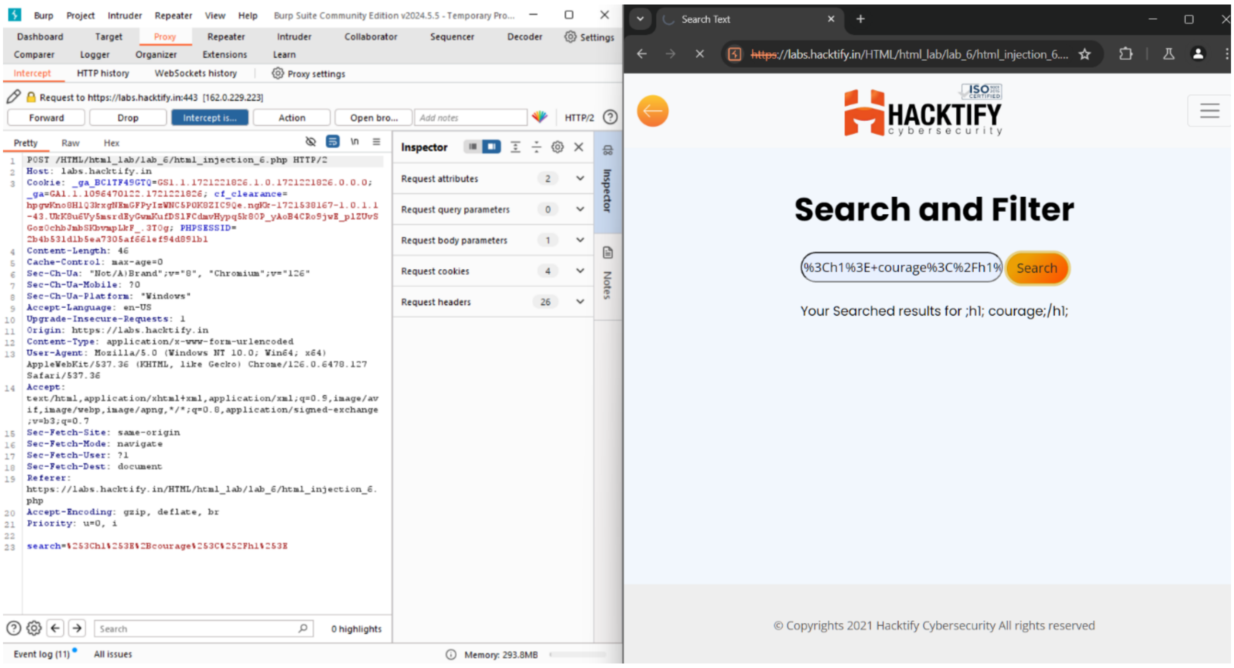
References

https://owasp.org/www-community/Injection_Information
https://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




Burp Suite Community Edition v2024.5.5 - Temporary Pro...
Dashboard Target Proxy Repeater Intruder Collaborator Sequencer Decoder Settings
Comparer Logger Organizer Extensions Learn
Intercept HTTP history WebSockets history Proxy settings
Forward Drop Intercept is on Action Open browser
Search Text
https://labs.hacktify.in/HTML/html_lab/lab_6/html_injection_6....
HACKTIFY cybersecurity
Search and Filter
%3Ch1%3E+courage%3C%2Fh1%
Search
Your Searched results for ;h1; courage;/h1;
© Copyrights 2021 Hacktify Cybersecurity All rights reserved


Burp Suite Community Edition v2024.5.5 - Temporary Pro...
Request to https://labs.hacktify.in:443 [162.0.229.223]
Forward Drop Intercept is... Action Open bro... HTTP/2
Pretty Raw Hex
POST /HTML/html_lab/lab_6/html_injection_6.php HTTP/2
Host: labs.hacktify.in
Content-Length: 46
Cache-Control: max-age=0
Origin: https://labs.hacktify.in
Content-Type: application/x-www-form-urlencoded
Referer: https://labs.hacktify.in/HTML/html_lab/lab_6/html_injection_6.php
Accept-Encoding: gzip, deflate, br
Priority: u=0, i
search=%253Ch1%253E%2Bcourage%253C%252Fh1%253E
Inspector
Request attributes 2
Request query parameters 0
Request body parameters 1
Request cookies 4
Request headers 26
Event log (11) All issues Memory: 293.8MB
Search Text
https://labs.hacktify.in/HTML/html_lab/lab_6/html_injection_6....
HACKTIFY cybersecurity
Search and Filter
%3Ch1%3E+courage%3C%2Fh1%
Search
Your Searched results for ;h1; courage;/h1;
© Copyrights 2021 Hacktify Cybersecurity All rights reserved

Proof of Concept