

Week 4

Penetration Testing Report

Introduction

This report document hereby describes the proceedings and results of a Black Box security assessment conducted against the **Week 4 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 4 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 Labs}, {Exchangeable Image File Format Labs}
------------------	---

3. Summary

Outlined is a Black Box Application Security assessment for the **Week 4 Labs**.

Total number of Sub-labs: 9 Sub-labs

High	Medium	Low
3	2	4

High - Number of Sub-labs with hard difficulty level

Medium - Number of Sub-labs with Medium difficulty level

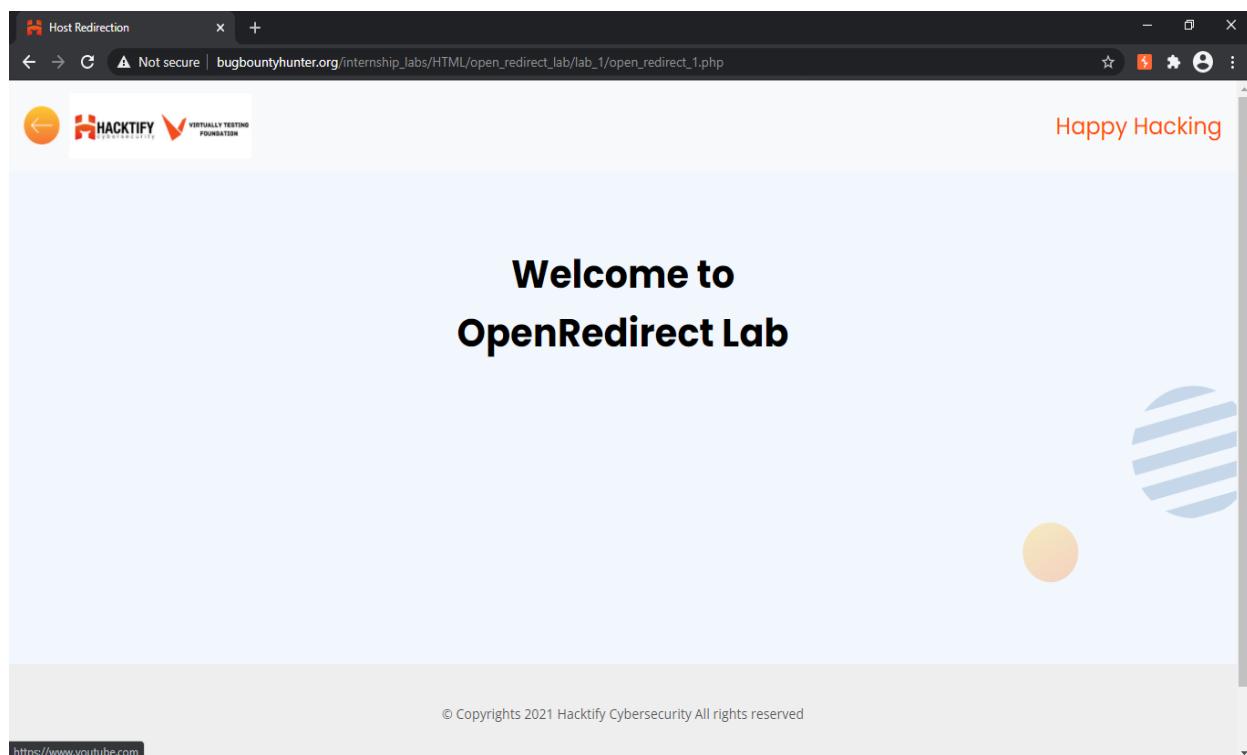
Low - Number of Sub-labs with Easy difficulty level

1. {Open Redirect Labs}

1.1. {A Simple Host!}

Reference	Risk Rating
A Simple Host!	Low
Tools Used	
Google Chrome and Burp Suite	
Vulnerability Description	
I found this vulnerability using the burp suite tool by intercepting it.	
How It Was Discovered	
Manual Analysis and Automated Analysis	
Vulnerable URLs	
https://www.bugbountyhunter.org/internship_labs/HTML/open_redirect_lab/lab_1/open_redirect_1.php	
Consequences of not Fixing the Issue	
An attacker can use this vulnerability to redirect users to other malicious websites, which can lead to phishing or other similar attacks.	
Suggested Countermeasures	
Maintain a server-side list of all URLs that are permitted for redirection. Instead of passing the target URL as a parameter to a redirector, pass an index into the list.	
References	
https://cheatsheetseries.owasp.org/cheatsheets/Unvalidated_Redirects_and_Forwards_Cheat_Sheet.html	

Proof of Concept



1.2. { Story of a beautiful header! }

Reference	Risk Rating
Story of a beautiful header	Low
Tools Used	
Google Chrome and Burp Suite	
Vulnerability Description	
I found this vulnerability by intercepting a redirect request into burp suite and inserting payload into it and then by forwarding through repeater by verifying the location of the payload result.	
How It Was Discovered	
Manual Analysis and Automated Analysis	
Vulnerable URLs	
https://www.bugbountyhunter.org/internship_labs/HTML/open_redirect_lab/lab_2/open_redirect_2.php	
Consequences of not Fixing the Issue	
An attacker can use this vulnerability to redirect users to other malicious websites, which can lead to phishing or other similar attacks.	
Suggested Countermeasures	
Remove the redirection function from the application, and replace links to it with direct links to the relevant target URLs.	
References	
https://cheatsheetseries.owasp.org/cheatsheets/Unvalidated_Redirects_and_Forwards_Cheat_Sheet.html	

Proof of Concept

Burp Suite Professional v2021.2.1 - Temporary Project - licensed to Uncia

Burp Project Intruder Repeater Window Help

Dashboard Target Proxy **Repeater** Sequencer Decoder Comparer Extender Project options User options

Send Cancel < > Follow redirection

Request

Pretty Raw In Actions ▾

```
1 GET /internship_lab_3/HTML/open_redirect_lab/lab_3/open_redirect_3.php?username=hacktify&
password=hacktify!kuuleyoutube.com/login>Login HTTP/1.1
2 Host: www.bugbountyhunter.org
3 Connection: close
4 Cache-Control: max-age=0
5 Sec-Fetch-Dest: frame
6 Sec-Fetch-Mode: "navigate"
7 Sec-Fetch-Site: "A Brand"
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/88.0.4324.150 Safari/537.36
9 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=0.9
10 Sec-Fetch-Site: sameorigin
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-User: ?1
13 Sec-Fetch-Dest: document
14 Accept-Encoding: gzip, deflate
15 Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
16 Cookie: PHPSESSID=opin7hbvnlikim57clughksa
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
```

Response

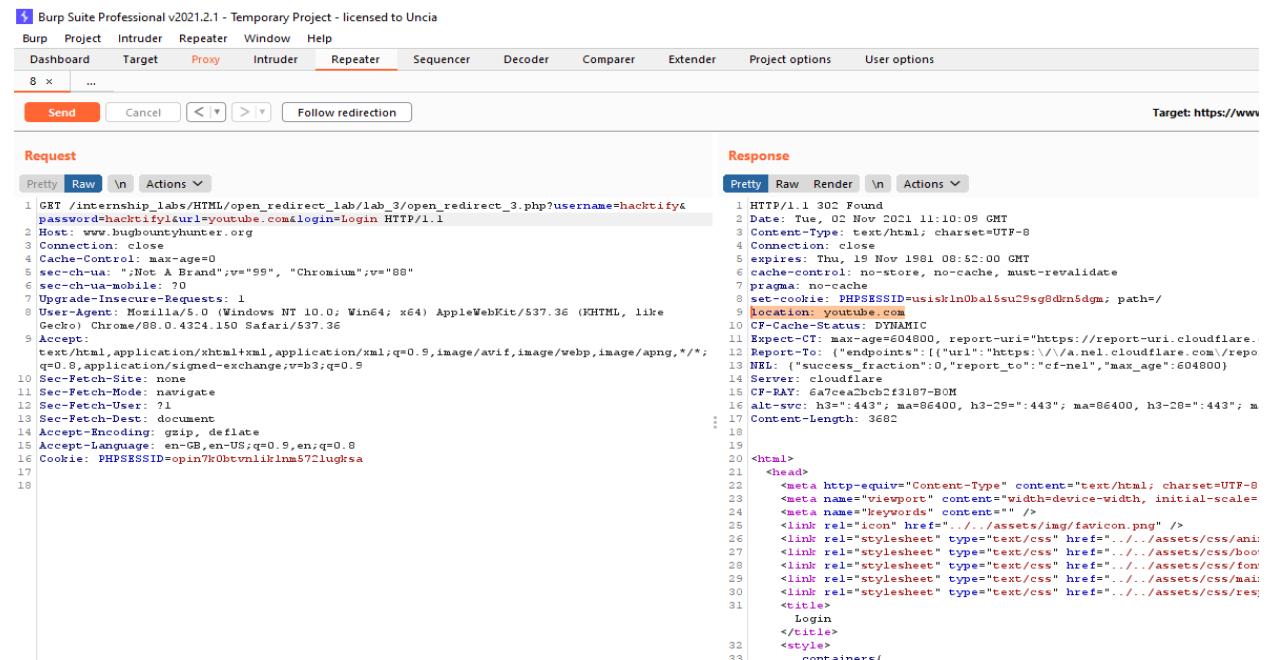
Pretty Raw Render In Actions ▾

```
1 HTTP/1.1 302 Found
2 Date: Tue, 02 Nov 2021 11:10:09 GMT
3 Content-Type: text/html; charset=UTF-8
4 Connection: close
5 expires: Thu, 19 Nov 1981 08:52:00 GMT
6 pragma: no-store, no-cache, must-revalidate
7 pragma: no-cache
8 set-cookie: PHPSESSID=usisklnDba5suC5sg8dn5dgm; path=/
9 location: youtube.com
10 CF-Cache-Status: DYNAMIC
11 Expect-CT: max-age=604800, report-uri="https://report-uri.cloudflare.com/cdn-cgi/beacon/e
NEL:{"name": "fraction": 0, "report_to": "cf-nel", "max_age": 604800}
12 Report-To: [{"endpoint": {"url": "https://\u2e2f.net.cloudflare.com/report/v3?e=Lc4pZyXAmU
NEL:{"name": "fraction": 0, "report_to": "cf-nel", "max_age": 604800}"}, "cf-ray": "fa7ccacbc2f187-BON
13 alt-svc: h3="443"; ma=64400, h3-25="443"; ma=64400, h3-26="443"; ma=64400, h3-27="443
14 Content-Length: 3692
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
```

1.3. {Sanitize Params!!}

Reference	Risk Rating
Sanitize params!!	Medium
Tools Used	
Google Chrome and Burp Suite	
Vulnerability Description	
I found this vulnerability by intercepting a redirect request into burp suite and inserting payload into it and then by forwarding through repeater by verifying the location of the payload result.	
How It Was Discovered	
Manual Analysis and Automated Analysis	
Vulnerable URLs	
https://www.bugbountyhunter.org/internship_labs/HTML/open_redirect_lab/lab_3/open_redirect_3_dashboard.php	
Consequences of not Fixing the Issue	
An attacker can use this vulnerability to redirect users to other malicious websites, which can lead to phishing or other similar attacks.	
Suggested Countermeasures	
The application should use relative URLs in all of its redirects, and the redirection function should strictly validate that the URL received is a valid URL.	
References	
https://cheatsheetseries.owasp.org/cheatsheets/Unvalidated_Redirects_and_Forwards_Cheat_Sheet.html	

Proof of Concept



The screenshot shows the Burp Suite Professional interface. The top navigation bar includes Project, Intruder, Repeater, Window, Help, Dashboard, Target, Proxy, Intruder, Repeater (which is selected), Sequencer, Decoder, Comparer, Extender, Project options, and User options. Below the navigation is a toolbar with Send, Cancel, and Follow redirection buttons. The status bar indicates the target is https://www.bugbountyhunter.org.

Request:

```
1 GET /internship_labs/HTML/open_redirect_lab/lab_3/open_redirect_3.php?username=hacktify&password=hacktify&url=youtube.com&login=Login HTTP/1.1
2 Host: www.bugbountyhunter.org
3 Connection: close
4 Cache-Control: max-age=0
5 sec-ch-ua: "Not A Brand";v="99", "Chromium";v="88"
6 sec-ch-ua-mobile: ?0
7 Upgrade-Insecure-Requests: 1
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4324.150 Safari/537.36
9 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8
10 Sec-Cookie: PHPSESSID=opin7k0btvnliknms72lughsa
11 Sec-Fetch-Site: none
12 Sec-Fetch-Mode: navigate
13 Sec-Fetch-User: ?1
14 Sec-Fetch-Dest: document
15 Accept-Encoding: gzip, deflate
16 Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
17 Cookie: PHPSESSID=opin7k0btvnliknms72lughsa
18
```

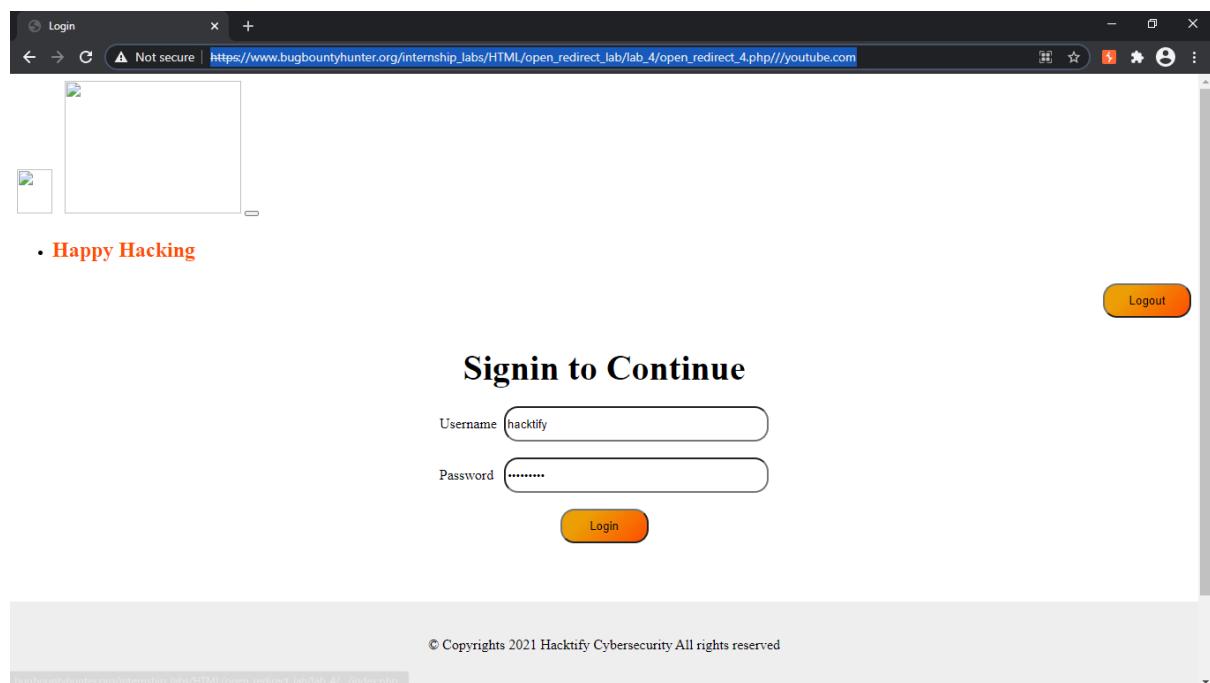
Response:

```
1 HTTP/1.1 302 Found
2 Date: Tue, 02 Nov 2021 11:10:09 GMT
3 Content-Type: text/html; charset=UTF-8
4 Connection: close
5 expires: Thu, 19 Nov 1981 08:52:00 GMT
6 cache-control: no-store, no-cache, must-revalidate
7 pragma: no-cache
8 set-cookie: PHPSESSID=usiskln0bal5su29sg@dkrn5dgm; path=/location/on/youtube.com
9 CF-Cache-Status: DYNAMIC
10 Expect-CT: max-age=604800, report-uri="https://report-uri.cloudflare.com/.../report-to/.../cf-nel", "max_ago": 604800
11 Report-To: {"endpoints": [{"url": "https://.../a.nel.cloudflare.com/.../nel": {"success_fraction": 0, "report_to": "cf-nel", "max_ago": 604800}}]
12 NEL: {"success_fraction": 0, "report_to": "cf-nel", "max_ago": 604800}
13 Server: cloudflare
14 CF-RAY: 6a7ceac6cb2f3187-BOM
15 alt-svc: h3=":443"; ma=86400, h3-29=:443; ma=86400, h3-28=:443; ma=86400
16 Content-Length: 3682
17
18
19
20 <html>
21   <head>
22     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"
23     <meta name="viewport" content="width=device-width, initial-scale=1"/>
24     <meta name="keywords" content="" />
25     <link rel="icon" href="../../assets/img/favicon.png" />
26     <link rel="stylesheet" type="text/css" href="../../assets/css/ani.css" />
27     <link rel="stylesheet" type="text/css" href="../../assets/css/bootstrap.css" />
28     <link rel="stylesheet" type="text/css" href="../../assets/css/font-awesome.css" />
29     <link rel="stylesheet" type="text/css" href="../../assets/css/main.css" />
30     <title> Login </title>
31     <style> .containers{
```

1.4. {Patterns are important!!}

Reference	Risk Rating
Patterns are important!!	Medium
Tools Used	
Google Chrome and Burp Suite	
Vulnerability Description	
I found this vulnerability by inserting payload at the end of link by adding "?" and then followed by a payload.	
How It Was Discovered	
Manual Analysis and Automated Analysis	
Vulnerable URLs	
//youtube.com">https://www.bugbountyhunter.org/internship_labs/HTML/open_redirect_lab/lab_4/open_redirect_4.php">//youtube.com	
Consequences of not Fixing the Issue	
An attacker can use this vulnerability to redirect users to other malicious websites, which can lead to phishing or other similar attacks.	
Suggested Countermeasures	
Maintain a server-side list of all URLs that are permitted for redirection. Instead of passing the target URL as a parameter to a redirector, pass an index into the list.	
References	
https://cheatsheetseries.owasp.org/cheatsheets/Unvalidated_Redirects_and_Forwards_Cheat_Sheet.html	

Proof of Concept



1.5. {File Upload? Redirect IT!}

Reference	Risk Rating
File upload? Redirect IT!	Low
Tools Used	
Google Chrome and Burp Suite	
Vulnerability Description	
I found this vulnerability by intercepting a file upload request into burp suite and changing file name to make it vulnerable.	
How It Was Discovered	
Manual Analysis and Automated Analysis	
Vulnerable URLs	
https://virtuallytesting.com/	
Consequences of not Fixing the Issue	
An attacker can use this vulnerability to redirect users to other malicious websites, which can lead to phishing or other similar attacks.	
Suggested Countermeasures	
Maintain a server-side list of all URLs that are permitted for redirection. Instead of passing the target URL as a parameter to a redirector, pass an index into the list.	
References	
https://cheatsheetseries.owasp.org/cheatsheets/Unvalidated_Redirects_and_Forwards_Cheat_Sheet.html	

Proof of Concept

```
Burp Suite Professional v2021.2.1 - Temporary Project - licensed to Uncia
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options
Intercept HTTP history WebSockets history Options
Request to https://virtuallytesting.com:443 [104.22.4.193]
Forward Drop Intercept is on Action Open Browser
Pretty Raw \n Actions ▾
1 GET / HTTP/1.1
2 Host: virtuallytesting.com
3 Connection: close
4 sec-ch-ua: ";Not A Brand";v="55", "Chromium";v="88"
5 sec-ch-ua-mobile: ?0
6 Upgrade-Insecure-Requests: 1
7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4324.150 Safari/537.36
8 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=0.9
9 Sec-Fetch-Site: cross-site
10 Sec-Fetch-Mode: navigate
11 Sec-Fetch-Dest: document
12 Referer: https://www.bugbountyhunter.org/
13 Accept-Encoding: gzip, deflate
14 Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
15
16
```

1.6. {Same Param Twice!}

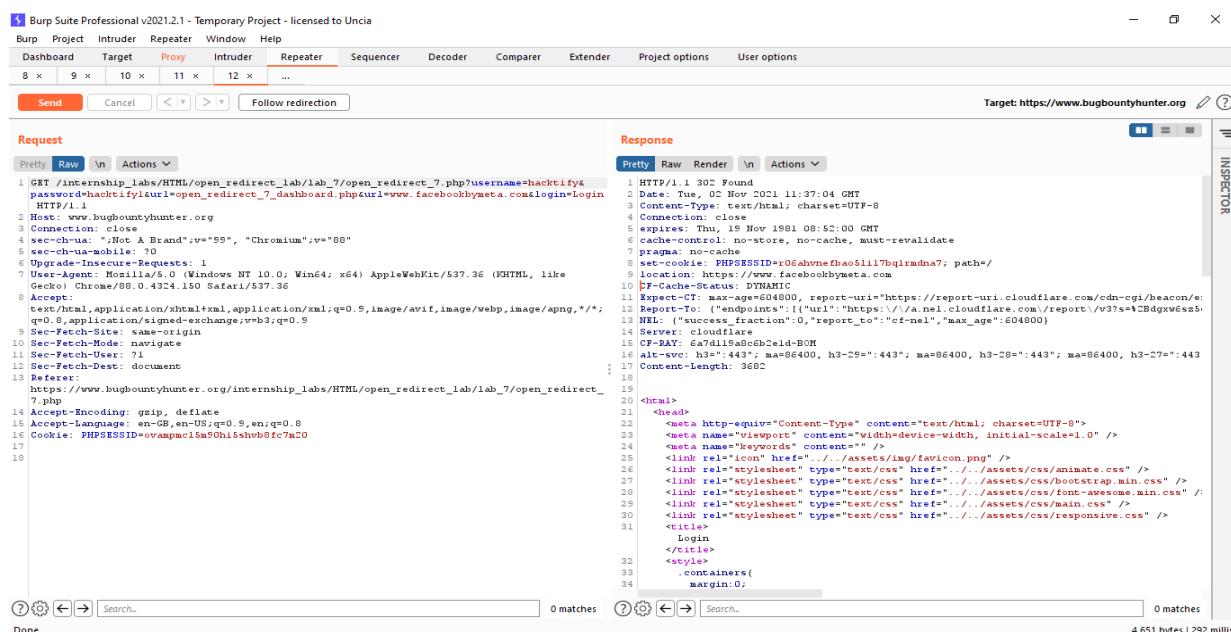
Reference	Risk Rating
Same Param Twice!	High
Tools Used	
Google Chrome and Burp Suite	
Vulnerability Description	
I found this vulnerability by intercepting a redirect request into burp suite and inserting payload into it and then by forwarding through repeater by verifying the location of the payload result.	
How It Was Discovered	
Manual Analysis and Automated Analysis	
Vulnerable URLs	
https://www.bugbountyhunter.org/internship_labs/HTML/open_redirect_lab/lab_6/open_redirect_6_dashboard.php	
Consequences of not Fixing the Issue	
An attacker can use this vulnerability to redirect users to other malicious websites, which can lead to phishing or other similar attacks.	
Suggested Countermeasures	
Remove the redirection function from the application, and replace links to it with direct links to the relevant target URLs.	
References	
https://cheatsheetseries.owasp.org/cheatsheets/Unvalidated_Redirects_and_Forwards_Cheat_Sheet.html	

Proof of Concept

1.7. {Domains? Not Always!}

Reference	Risk Rating
Domains? Not Always!	High
Tools Used	
Google Chrome and Burp Suite	
Vulnerability Description	
I found this vulnerability by intercepting a redirect request into burp suite and inserting payload into it and then by forwarding through repeater by verifying the location of the payload result.	
How It Was Discovered	
Manual Analysis and Automated Analysis	
Vulnerable URLs	
https://www.bugbountyhunter.org/internship_labs/HTML/open_redirect_lab/lab_7/open_redirect_7_dashboard.php?url=google.com&login=Login	
Consequences of not Fixing the Issue	
An attacker can use this vulnerability to redirect users to other malicious websites, which can lead to phishing or other similar attacks.	
Suggested Countermeasures	
Maintain a server-side list of all URLs that are permitted for redirection. Instead of passing the target URL as a parameter to a redirector, pass an index into the list.	
References	
https://cheatsheetseries.owasp.org/cheatsheets/Unvalidated_Redirects_and_Forwards_Cheat_Sheet.html	

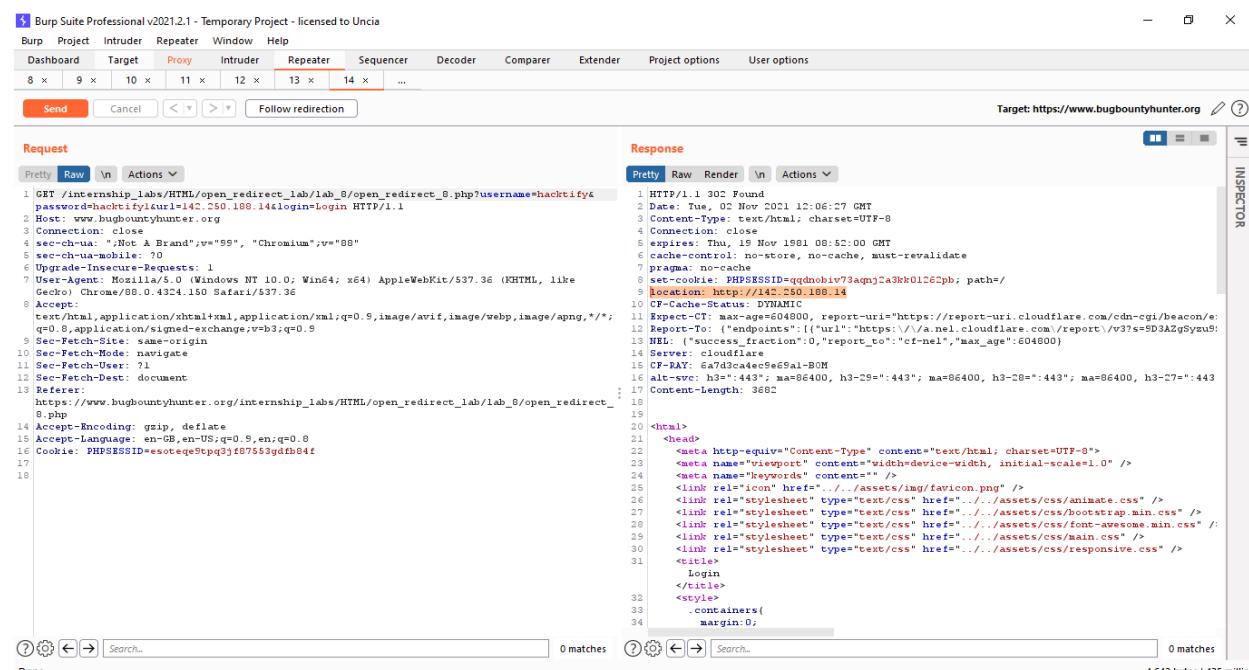
Proof of Concept



1.8. {Style digit symbols < 3}

Reference	Risk Rating
Style digit symbols < 3	High
Tools Used	
Google Chrome and Burp Suite	
Vulnerability Description	
I found this vulnerability by intercepting a redirect request into burp suite and inserting IP Address as payload into it and then by forwarding through repeater by verifying the location of the payload result.	
How It Was Discovered	
Manual Analysis and Automated Analysis	
Vulnerable URLs	
https://www.bugbountyhunter.org/internship_labs/HTML/open_redirect_lab/lab_8/open_redirect_8.php	
https://www.google.com/?gws_rd=ssl	
Consequences of not Fixing the Issue	
An attacker can use this vulnerability to redirect users to other malicious websites, which can lead to phishing or other similar attacks.	
Suggested Countermeasures	
Remove the redirection function from the application, and replace links to it with direct links to the relevant target URLs.	
References	
https://cheatsheetseries.owasp.org/cheatsheets/Unvalidated_Redirects_and_Forwards_Cheat_Sheet.html	

Proof of Concept



The screenshot shows the Burp Suite Professional interface. The 'Proxy' tab is selected, displaying a captured request and response. The request is a GET to https://www.bugbountyhunter.org/internship_labs/HTML/open_redirect_lab/lab_8/open_redirect_8.php?username=hacktify&password=hacktify&lurl=142.250.180.14&login>Login. The response shows a 302 Found status with a Location header pointing to http://142.250.186.14. The response body contains a large amount of HTML and CSS code, indicating a successful exploit of the vulnerability.

```
HTTP/1.1 302 Found
Date: Tue, 02 Nov 2021 12:06:27 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Set-Cookie: PHPSESSID=qdnobnbiw73agnj2a3kk012e2pb; path=/; location: http://142.250.186.14
CF-Cache-Status: DYNAMIC
Expect-CT: max-age=604800, report-uri="https://report-uri.cloudflare.com/cdn-cgi/beacon/e:report-to=({"endpoints": [{"url": "https://v3.sqd3azg5yus9.net:443", "fraction": 0}, {"url": "cf-nel", "max_age": 604800}], "nel": {"max_size": 1024, "interval": 1000}}, "max-age=604800"
Report-To: ("endpoints": [{"url": "https://v3.sqd3azg5yus9.net:443", "fraction": 0}, {"url": "cf-nel", "max_age": 604800}], "nel": {"max_size": 1024, "interval": 1000})
Server: cloudflare
Alt-PAY: 6a7d3ca44a5e69a1-BOM
Alt-SVC: h3=":443"; ma=86400, h3-29=:443; ma=86400, h3-28=:443; ma=86400, h3-27=:443
Content-Length: 3682
Content-Type: text/html; charset=UTF-8
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <meta name="keywords" content="" />
    <link rel="icon" href="/assets/img/favicon.png" />
    <link rel="stylesheet" type="text/css" href="/assets/css/animate.css" />
    <link rel="stylesheet" type="text/css" href="/assets/css/bootstrap.min.css" />
    <link rel="stylesheet" type="text/css" href="/assets/css/font-awesome.min.css" />
    <link rel="stylesheet" type="text/css" href="/assets/css/main.css" />
    <link rel="stylesheet" type="text/css" href="/assets/css/responsive.css" />
  </head>
  <body>
    <div class="container" style="text-align: center; margin-top: 100px;">
      <h1>Login</h1>
      <form>
        <div class="form-group">
          <label for="username">Username</label>
          <input type="text" id="username" name="username" value="h33llo" />
        </div>
        <div class="form-group">
          <label for="password">Password</label>
          <input type="password" id="password" name="password" value="h33llo" />
        </div>
        <div class="checkbox">
          <label for="remember-me">Remember Me</label>
          <input type="checkbox" id="remember-me" name="remember-me" checked="" checked="" />
        </div>
        <div class="form-group">
          <button type="submit" class="btn btn-primary" value="Submit" style="width: 100%; height: 40px; font-size: 16px; background-color: #337ab7; color: white; border: none; border-radius: 5px; padding: 10px 0; font-weight: bold; text-decoration: none; transition: all 0.3s ease; text-align: center; margin-top: 10px;">Submit</button>
        </div>
      </form>
    </div>
  </body>
</html>
```

2. {Exchangeable Image File Format Labs}

2.1. Lets PII!

Reference	Risk Rating
Lets PII!	Low
Tools Used	
Google Chrome	
Vulnerability Description	
I found this vulnerability by uploading an image into the profile photo section. I found the entry point for EXIF in the profile photo upload section. This allowed me to upload images containing sensitive data.	
How It Was Discovered	
Manual Analysis	
Vulnerable URLs	
https://www.bugbountyhunter.org/internship_labs/HTML/exif_lab/lab_1/exif.php	
Consequences of not Fixing the Issue	
This vulnerability violates the privacy of a user and shares sensitive information of the user who uploaded an image on the vulnerable websites.	
Suggested Countermeasures	
Disable geotagging on the digital device you use to take photographs.	
References	
https://photographylife.com/what-is-exif-data	

Proof of Concept

