

# Week 6 Penetration Testing Report

## Introduction

This report document hereby describes the proceedings and results of a Black Box security assessment conducted against the **Week 6 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 6 Labs** and provide a final security assessment report comprising vulnerabilities, remediation strategy and recommendation guidelines to help mitigate the identified vulnerabilities and risks during the activity.

## 2. Scope

This section defines the scope and boundaries of the project.

Application Name	{Cross-Site Request Forgery}
------------------	------------------------------

## 3. Summary

Outlined is a Black Box Application Security assessment for the **Week 6 Labs**.

**Total number of Sub-labs: 6 Sub-labs**

High	Medium	Low
2	2	2

**High** - Number of Sub-labs with hard difficulty level

**Medium** - Number of Sub-labs with Medium difficulty level

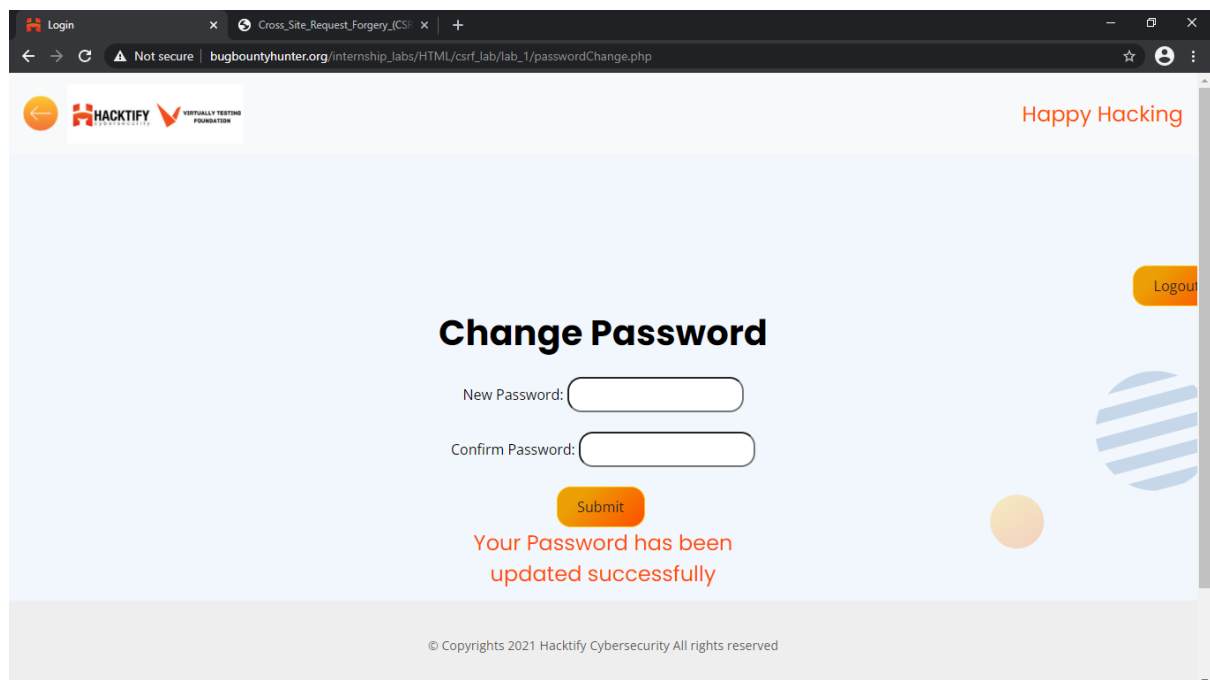
**Low** - Number of Sub-labs with Easy difficulty level

# 1. {Cross-Site Request Forgery}

## 1.1. {Easy CSRF}

Reference	Risk Rating
Easy CSRF	Low
<b>Tools Used</b>	
Google Chrome, CSRF tool, Burp Suite	
<b>Vulnerability Description</b>	
I found this vulnerability by intercepting a password changing request through an attacker account and forged it to get access to the victims account. Then I send this request to the victim and when the victim opens and clicks on the request, the password will change successfully and will get access to the victim's account.	
<b>How It Was Discovered</b>	
Automated Tools and Manual Analysis	
<b>Vulnerable URLs</b>	
<a href="https://www.bugbountyhunter.org/internship_labs/HTML/csrf_lab/lab_1/passwordChange.php">https://www.bugbountyhunter.org/internship_labs/HTML/csrf_lab/lab_1/passwordChange.php</a>	
<b>Consequences of not Fixing the Issue</b>	
Attacker causes the victim user to carry out unintentional actions like taking over users accounts illegally. Compromising privileges role within the application, and taking over applications data and functions.	
<b>Suggested Countermeasures</b>	
Anti CSRF tokens, different for different users, and Same site cookies.	
<b>References</b>	
<a href="https://owasp.org/www-community/attacks/csrf">https://owasp.org/www-community/attacks/csrf</a>	

## Proof of Concept



## 1.2. {Always Validate Tokens}

Reference	Risk Rating
Always Validate Tokens	Medium
<b>Tools Used</b>	
Google Chrome, CSRF tool, Burp Suite	
<b>Vulnerability Description</b>	
I found this vulnerability by intercepting a password changing request through an attacker account and forged it to get access to the victims account. Then I send this request to the victim and when the victim opens and clicks on the request, the password will change successfully and will get access to the victim's account.	
<b>How It Was Discovered</b>	
Automated Tools and Manual Analysis	
<b>Vulnerable URLs</b>	
<a href="https://www.bugbountyhunter.org/internship_labs/HTML/csrf_lab/lab_2/passwordChange.php">https://www.bugbountyhunter.org/internship_labs/HTML/csrf_lab/lab_2/passwordChange.php</a>	
<b>Consequences of not Fixing the Issue</b>	
Attacker causes the victim user to carry out unintentional actions like taking over users accounts illegally. Compromising privileges role within the application, and taking over applications data and functions.	
<b>Suggested Countermeasures</b>	
Anti CSRF tokens, different for different users, and Same site cookies.	
<b>References</b>	
<a href="https://www.acunetix.com/websitesecurity/csrf-attacks/">https://www.acunetix.com/websitesecurity/csrf-attacks/</a>	

## Proof of Concept

The screenshot shows a web browser window with the URL [https://www.bugbountyhunter.org/internship\\_labs/HTML/csrf\\_lab/lab\\_2/passwordChange.php](https://www.bugbountyhunter.org/internship_labs/HTML/csrf_lab/lab_2/passwordChange.php). The page displays a 'Change Password' form with two input fields: 'New Password:' and 'Confirm Password:'. Below the form is a 'Submit' button. A message below the form states 'Your Password has been updated successfully'. The footer of the page reads '© Copyrights 2021 Hacktify Cybersecurity All rights reserved'.

Overlaid on the right side of the browser window is a Burp Suite window titled 'CSRF PoC generator'. The 'Request to:' field shows the same URL as the browser. The 'Raw' tab is selected, displaying a list of request details. The 'CSRF HTML' tab is also visible, showing a generated HTML payload for a CSRF attack. The payload includes a script to push the state and a form action to submit the password change request.

## 1.3. {I Hate When Someone Uses My Tokens!}

Reference	Risk Rating
I Hate When Someone Uses My Tokens!	Medium
<b>Tools Used</b>	
Google Chrome, CSRF tool, Burp Suite	
<b>Vulnerability Description</b>	
I found this vulnerability by intercepting a password changing request through an attacker account and forged it to get access to the victims account. Then I send this request to the victim and when the victim opens and clicks on the request, the password will change successfully and will get access to the victim's account.	
<b>How It Was Discovered</b>	
Automated Tools and Manual Analysis	
<b>Vulnerable URLs</b>	
<a href="https://www.bugbountyhunter.org/internship_labs/HTML/csrf_lab/lab_4/passwordChange.php">https://www.bugbountyhunter.org/internship_labs/HTML/csrf_lab/lab_4/passwordChange.php</a>	
<b>Consequences of not Fixing the Issue</b>	
Attacker causes the victim user to carry out unintentional actions like taking over users accounts illegally. Compromising privileges role within the application, and taking over applications data and functions.	
<b>Suggested Countermeasures</b>	
Anti CSRF tokens, different for different users, and Same site cookies.	
<b>References</b>	
<a href="https://portswigger.net/web-security/csrf">https://portswigger.net/web-security/csrf</a>	

## Proof of Concept

The screenshot displays the 'CSRF PoC generator' interface. At the top, the request URL is set to `https://www.bugbountyhunter.org`. The 'Raw' tab is selected, showing the raw HTTP request details. The 'Inspector' panel on the right shows the 'Accept' header. Below the request details, the 'CSRF HTML' section shows the generated HTML payload, which includes a form with hidden fields for 'newPassword2', 'csrf', and a submit button.

```
1 POST /internship_labs/HTML/csrf_lab/lab_4/passwordChange.php HTTP/1.1
2 Host: www.bugbountyhunter.org
3 Connection: close
4 Content-Length: 74
5 Cache-Control: max-age=0
6 sec-ch-ua: ";Not A Brand";v="99", "Chromium";v="88"
7 sec-ch-ua-mobile: ?0
8 Upgrade-Insecure-Requests: 1
9 Origin: https://www.bugbountyhunter.org
10 Content-Type: application/x-www-form-urlencoded
11 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
12 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
```

CSRF HTML:

```
1 <html>
2 <!-- CSRF PoC - generated by Burp Suite Professional -->
3 <body>
4 <script>history.pushState('', '', '/')</script>
5 <form action="https://www.bugbountyhunter.org/internship_labs/HTML/csrf_lab/lab_4/passwordChange.php" method="POST">
6 <input type="hidden" name="newPassword" value="bunny" />
7 <input type="hidden" name="newPassword2" value="bunny" />
8 <input type="hidden" name="csrf" value="9e6dc8685bf3c1b338f2011ace904887" />
9 <input type="submit" value="Submit request" />
10 </form>
11 </body>
12 </html>
```

## 1.4. {GET Me or POST ME}

Reference	Risk Rating
GET Me or POST ME	Low
<b>Tools Used</b>	
Google Chrome, CSRF tool, Burp Suite	
<b>Vulnerability Description</b>	
I found this vulnerability by intercepting a password changing request through an attacker account and forged it to get access to the victims account. Then I send this request to the victim and when the victim opens and clicks on the request, the password will change successfully and will get access to the victim's account.	
<b>How It Was Discovered</b>	
Automated Tools and Manual Analysis	
<b>Vulnerable URLs</b>	
<a href="https://www.bugbountyhunter.org/internship_labs/HTML/csrf_lab/lab_6/passwordChange.php?newPassword=bunny&amp;newPassword2=bunny&amp;csrf=9f30abfb7a0141bb657fa6d587a5878b">https://www.bugbountyhunter.org/internship_labs/HTML/csrf_lab/lab_6/passwordChange.php?newPassword=bunny&amp;newPassword2=bunny&amp;csrf=9f30abfb7a0141bb657fa6d587a5878b</a>	
<b>Consequences of not Fixing the Issue</b>	
Attacker causes the victim user to carry out unintentional actions like taking over users accounts illegally. Compromising privileges role within the application, and taking over applications data and functions.	
<b>Suggested Countermeasures</b>	
Anti CSRF tokens, different for different users, and Same site cookies.	
<b>References</b>	
<a href="https://owasp.org/www-community/attacks/csrf">https://owasp.org/www-community/attacks/csrf</a>	

## Proof of Concept

The screenshot displays the Burp Suite interface. The top section shows a raw HTTP POST request to `/internship_labs/HTML/csrf_lab/lab_6/passwordChange.php`. The request headers include `Host: www.bugbountyhunter.org`, `Connection: close`, `Content-Length: 70`, `Cache-Control: max-age=0`, `sec-ch-ua: ";Not A Brand";v="99", "Chromium";v="88"`, `sec-ch-ua-mobile: ?0`, `Upgrade-Insecure-Requests: 1`, `Origin: https://www.bugbountyhunter.org`, `Content-Type: application/x-www-form-urlencoded`, `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`, and `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`. The `Sec-Fetch-Site` header is set to `same-origin`.

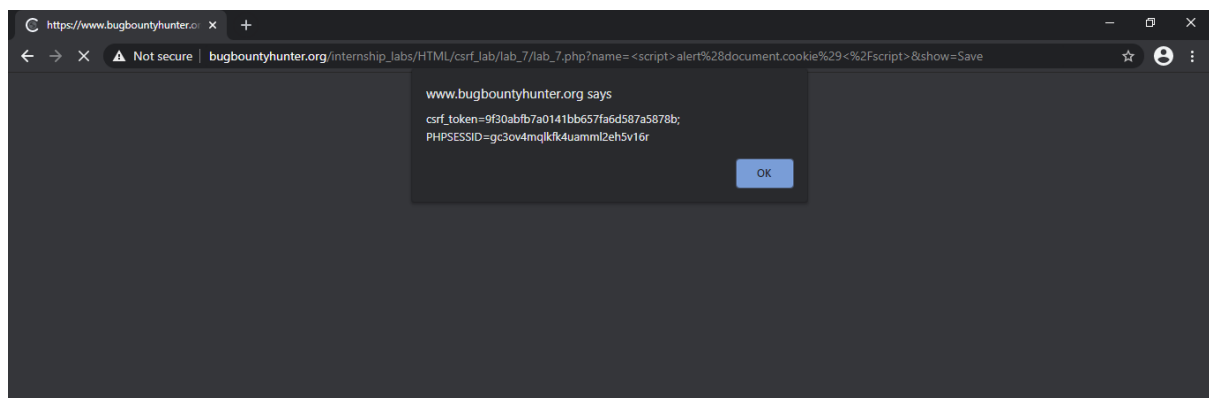
The bottom section shows the generated CSRF HTML payload, which is a form with the following fields:

```
<form action="https://www.bugbountyhunter.org/internship_labs/HTML/csrf_lab/lab_6/passwordChange.php" method="GET">
  <input type="hidden" name="newPassword" value="bunny" />
  <input type="hidden" name="newPassword2" value="bunny" />
  <input type="hidden" name="csrf" value="9f30abfb7a0141bb657fa6d587a5878b" />
  <input type="submit" value="Submit request" />
</form>
```

## 1.5. {XSS is Saviour}

Reference	Risk Rating
XSS is Saviour	Hard
<b>Tools Used</b>	
Google Chrome, CSRF tool, Burp Suite	
<b>Vulnerability Description</b>	
I found this vulnerability by intercepting a password changing request through an attacker account and forged it to get access to the victims account. Then I send this request to the victim and when the victim opens and clicks on the request, the password will change successfully and will get access to the victim's account.	
<b>How It Was Discovered</b>	
Automated Tools and Manual Analysis	
<b>Vulnerable URLs</b>	
<a href="https://www.bugbountyhunter.org/internship_labs/HTML/csrf_lab/lab_7/lab_7.php?name=%3Cscript%3Ealert%28document.cookie%29%3C%2Fscript%3E&amp;show=Save">https://www.bugbountyhunter.org/internship_labs/HTML/csrf_lab/lab_7/lab_7.php?name=%3Cscript%3Ealert%28document.cookie%29%3C%2Fscript%3E&amp;show=Save</a>	
<b>Consequences of not Fixing the Issue</b>	
Attacker causes the victim user to carry out unintentional actions like taking over users accounts illegally. Compromising privileges role within the application, and taking over applications data and functions.	
<b>Suggested Countermeasures</b>	
Anti CSRF tokens, different for different users, and Same site cookies.	
<b>References</b>	
<a href="https://www.acunetix.com/websitesecurity/csrf-attacks/">https://www.acunetix.com/websitesecurity/csrf-attacks/</a>	

## Proof of Concept



## 1.6. {Rm - Rf Token}

Reference	Risk Rating
Rm - Rf Token	Hard
Tools Used	
Google Chrome, CSRF tool, Burp Suite	
Vulnerability Description	
I found this vulnerability by intercepting a password changing request through an attacker account and forged it to get access to the victims account. Then I send this request to the victim and when the victim opens and clicks on the request, the password will change successfully and will get access to the victim's account.	
How It Was Discovered	
Automated Tools and Manual Analysis	
Vulnerable URLs	
<a href="https://www.bugbountyhunter.org/internship_labs/HTML/csrf_lab/lab_8/passwordChange.php?newPassword=bunny&amp;newPassword2=bunny&amp;csrf=9f30abfb7a0141bb657fa6d587a5878b">https://www.bugbountyhunter.org/internship_labs/HTML/csrf_lab/lab_8/passwordChange.php?newPassword=bunny&amp;newPassword2=bunny&amp;csrf=9f30abfb7a0141bb657fa6d587a5878b</a>	
Consequences of not Fixing the Issue	
Attacker causes the victim user to carry out unintentional actions like taking over users accounts illegally. Compromising privileges role within the application, and taking over applications data and functions.	
Suggested Countermeasures	
Anti CSRF tokens, different for different users, and Same site cookies.	
References	
<a href="https://portswigger.net/web-security/csrf">https://portswigger.net/web-security/csrf</a>	

## Proof of Concept

The image shows a web browser window displaying a 'Change Password' form on the website [https://www.bugbountyhunter.org/internship\\_labs/HTML/csrf\\_lab/lab\\_8/](https://www.bugbountyhunter.org/internship_labs/HTML/csrf_lab/lab_8/). The form has two input fields: 'New Password:' and 'Confirm Password:', followed by a 'Submit' button. Below the form, a message states: 'Your Password has been updated successfully'. At the bottom of the page, it says '© Copyrights 2021 Hacktify Cybersecurity All rights reserved'.

Overlaid on the right side of the browser window is the 'CSRF PoC generator' tool. The 'Request to:' field is set to [https://www.bugbountyhunter.org/internship\\_labs/HTML/csrf\\_lab/lab\\_8/passwordChange.php](https://www.bugbountyhunter.org/internship_labs/HTML/csrf_lab/lab_8/passwordChange.php). The tool shows a list of request headers and body parameters. The 'Referer' header is set to 'rm -rf'. The 'CSRF HTML' section shows the generated HTML code for the PoC, which includes a form with hidden fields for 'newPassword' and 'newPassword2', and a submit button.