Week 3 Penetration Testing Report

Introduction

This report document hereby describes the proceedings and results of a Black Box security assessment conducted against the **Week 3 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 3 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 Scripting}**

3. Summary

Outlined is a Black Box Application Security assessment for the **Week 3 Labs**.

Total number of Sub-labs: 11 Sub-labs

High	Medium	Low
3	3	5

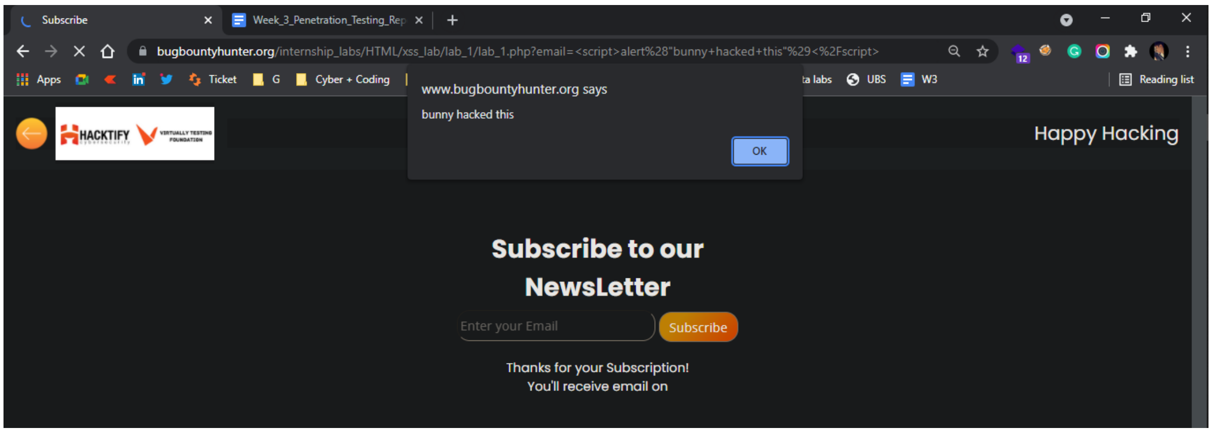
High	-	**Number of Sub-labs with hard difficulty level**
Medium	-	**Number of Sub-labs with Medium difficulty level**
Low	-	**Number of Sub-labs with Easy difficulty level**

1. {Cross-Site Scripting}

1.1. {Let's do it!}

Reference	**Risk Rating**
Let's Do IT!	**Low**
Tools Used	
Google Chrome	
Vulnerability Description	
I found this vulnerability by entering simple javascript code into the client side reflection field, i.e., i successfully performed reflected xss.	
How It Was Discovered	
Manual Analysis	
Vulnerable URLs	
https://www.bugbountyhunter.org/internship_labs/HTML/xss_lab/lab_1/lab_1.php?email=%3Cscript%3Ealert%28%22bunny+hacked+this%22%29%3C%2Fscript%3E	
Consequences of not Fixing the Issue	
XSS enables an attacker to inject client side code, so with this, an attacker will be able to capture client side cookies which can capture account credentials of any user.	
Suggested Countermeasures	
Developers must sanitize the input of any user by denying entry of any tag or any special character.	
References	
I found this basic vulnerability by entering a basic JavaScript tag. So, I did not take any help.	

Proof of Concept


bugbountyhunter.org/internship_labs/HTML/xss_lab/lab_1/lab_1.php?email=<script>alert%28"bunny+hacked+this"%29<%2Fscript>

www.bugbountyhunter.org says

bunny hacked this

OK

Happy Hacking

Subscribe to our NewsLetter

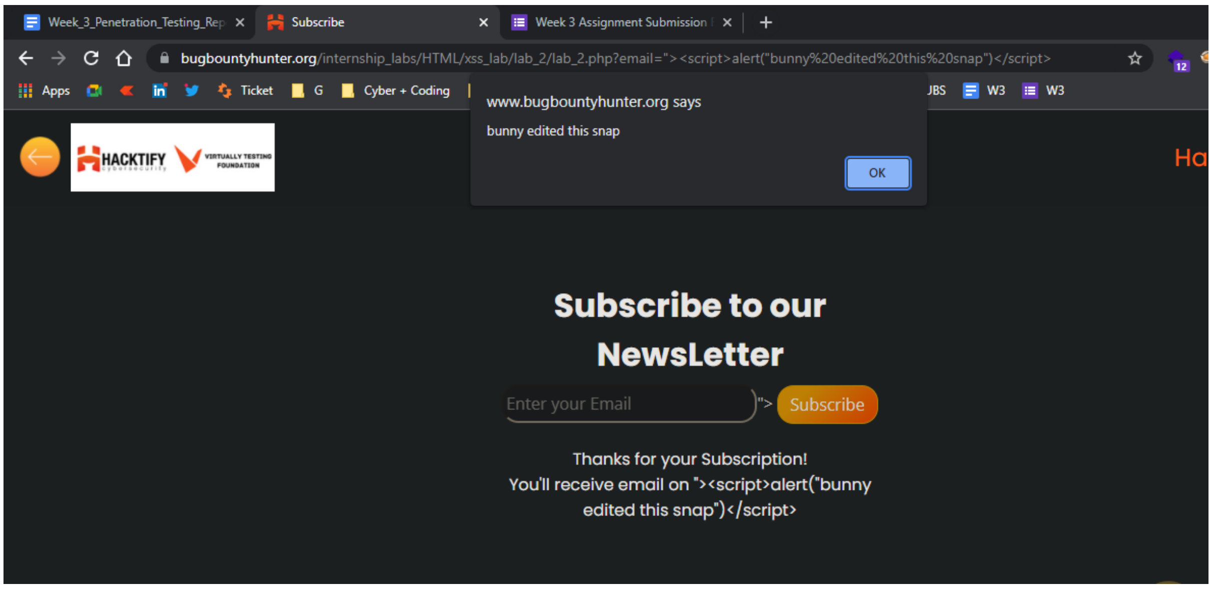
Enter your Email | Subscribe

Thanks for your Subscription!
You'll receive email on

1.2. {Balancing is important in life}

Reference	**Risk Rating**
Balancing is important in life	**Low**
Tools Used	
Google chrome	
Vulnerability Description	
I found this vulnerability by modifying the URL after getting valid output as actually developed. Also, it will allow an attacker to carry out any action that the user is able to perform.	
How It Was Discovered	
Manual Analysis	
Vulnerable URLs	
https://www.bugbountyhunter.org/internship_labs/HTML/xss_lab/lab_2/lab_2.php?email=%22%3E%3Cscript%3Ealert(%22bunny%20edited%20this%20snap%22)%3C/script%3E	
Consequences of not Fixing the Issue	
Carry out any action that the user is able to perform.	
Suggested Countermeasures	
Developers must sanitize the input of any user by denying entry of any tag or any special character. And also illegal links of any web page should not display any valid output.	
References	
https://github.com/payloadbox/xss-payload-list	

Proof of Concept



```
bugbountyhunter.org/internship_labs/HTML/xss_lab/lab_2/lab_2.php?email="><script>alert("bunny%20edited%20this%20snap")</script>
```

www.bugbountyhunter.org says

bunny edited this snap

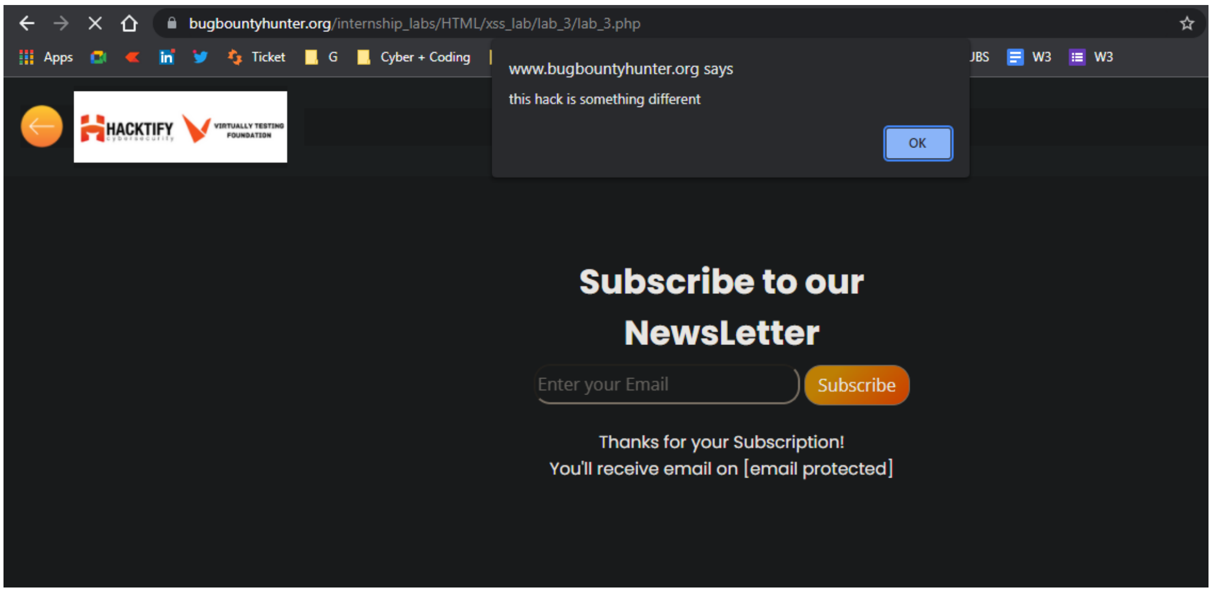
Subscribe to our NewsLetter

Thanks for your Subscription!
You'll receive email on "><script>alert("bunny edited this snap")</script>

1.3. {XSS is everywhere!}

Reference	Risk Rating
XSS is everywhere!	**Low**
Tools Used	
Google chrome	
Vulnerability Description	
I found this vulnerability by modifying the URL after getting valid output as actually developed. Here, the developer has given validation while developing this web page as the text field must contain a valid email address which will end with "@gmail.com". If any other input is entered, then an error message will display called "Enter a valid email address". So, to bypass this I simple inserted my payload which will end with provided validation for input, i.e., "@gmail.com"	
How It Was Discovered	
Manual Analysis	
Vulnerable URLs	
https://www.bugbountyhunter.org/internship_labs/HTML/xss_lab/lab_3/lab_3.php	
Consequences of not Fixing the Issue	
Carry out any action that the user is able to perform.	
Suggested Countermeasures	
Developers must sanitize the input of any user by denying entry of any tag or any special character if it is not a valid entry for that particular text field.	
References	
https://github.com/payloadbox/xss-payload-list	

Proof of Concept


bugbountyhunter.org/internship_labs/HTML/xss_lab/lab_3/lab_3.php

www.bugbountyhunter.org says

this hack is something different

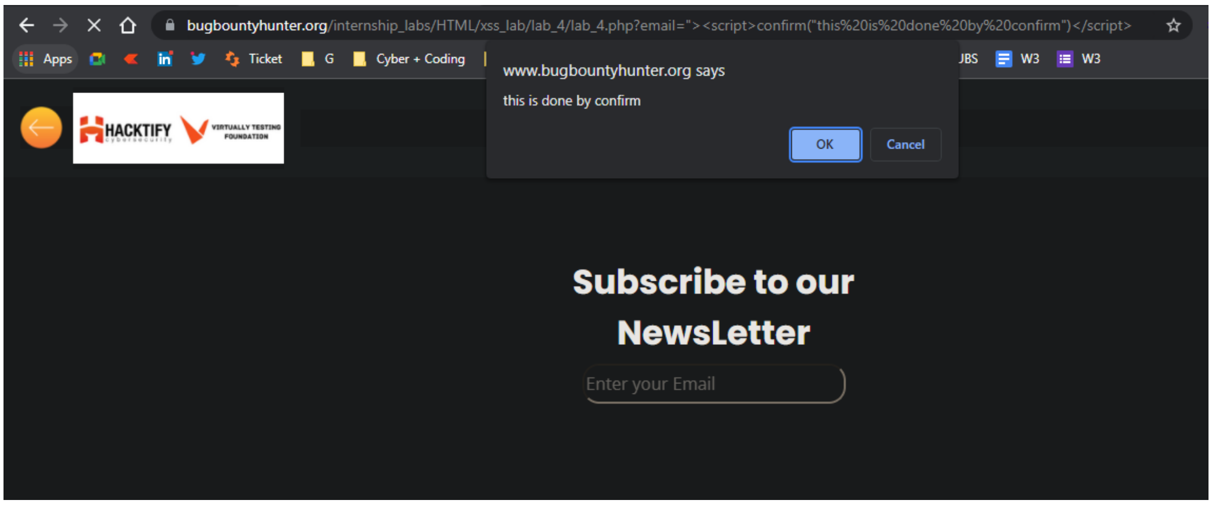
Subscribe to our NewsLetter

Thanks for your Subscription!
You'll receive email on [email protected]

1.4. {Alternatives are a must!}

Reference	**Risk Rating**
Alternatives are a must	**Medium**
Tools Used	
Google chrome	
Vulnerability Description	
I found this vulnerability by modifying the URL after getting valid output as actually developed. But, here the developer has sanitized the tag which is commonly used in JavaScript, i.e., alert(). So, to bypass this situation, I used the confirm() tag and I successfully got what I expected. Also, I am able to successfully run prompt() tag instead of alert() and alternative of confirm()	
How It Was Discovered	
Manual Analysis	
Vulnerable URLs	
https://www.bugbountyhunter.org/internship_labs/HTML/xss_lab/lab_4/lab_4.php?email=%22%3E%3Cscript%3Econfirm(%22this%20is%20done%20by%20confirm%22)%3C/script%3E	
Consequences of not Fixing the Issue	
Carry out any action that the user is able to perform.	
Suggested Countermeasures	
Such web pages must be sanitized by denying input of JavaScript tags which are not required in the text fields.	
References	
https://github.com/payloadbox/xss-payload-list	

Proof of Concept



bugbountyhunter.org/internship_labs/HTML/xss_lab/lab_4/lab_4.php?email="><script>confirm("this%20is%20done%20by%20confirm")</script>

www.bugbountyhunter.org says

this is done by confirm

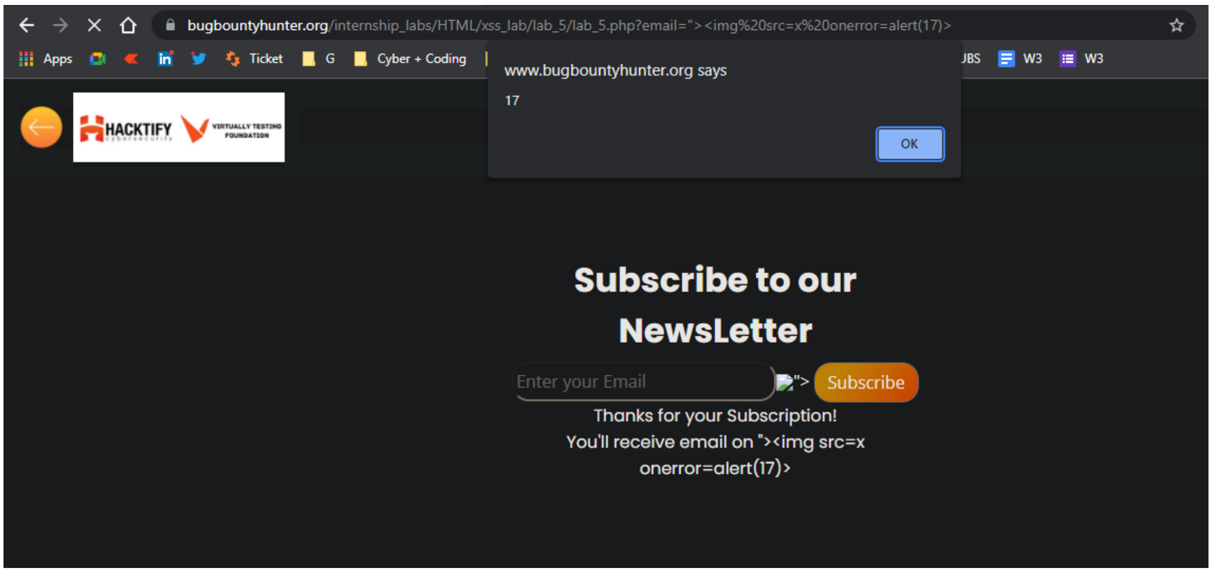
Subscribe to our NewsLetter

Enter your Email

1.5. {Developer hates script!}

Reference	**Risk Rating**
Developers hate scripts!	**Hard**
Tools Used	
Google chrome	
Vulnerability Description	
I found this vulnerability by using different javascript tags like <img>, <svg>, etc.	
How It Was Discovered	
Manual Analysis	
Vulnerable URLs	
https://www.bugbountyhunter.org/internship_labs/HTML/xss_lab/lab_5/lab_5.php?email=%22%3E%3Cimg%20src=x%20onerror=alert(17)%3E	
Consequences of not Fixing the Issue	
Attackers will be able to read the data that the user is able to access.	
Suggested Countermeasures	
Developers must sanitize the input of any user by denying entry of any tag or any special character if it is not a valid entry for that particular text field.	
References	
https://github.com/payloadbox/xss-payload-list	

Proof of Concept


bugbountyhunter.org/internship_labs/HTML/xss_lab/lab_5/lab_5.php?email="><img%20src=x%20onerror=alert(17)>

www.bugbountyhunter.org says

17

OK

Subscribe to our NewsLetter

Enter your Email ">

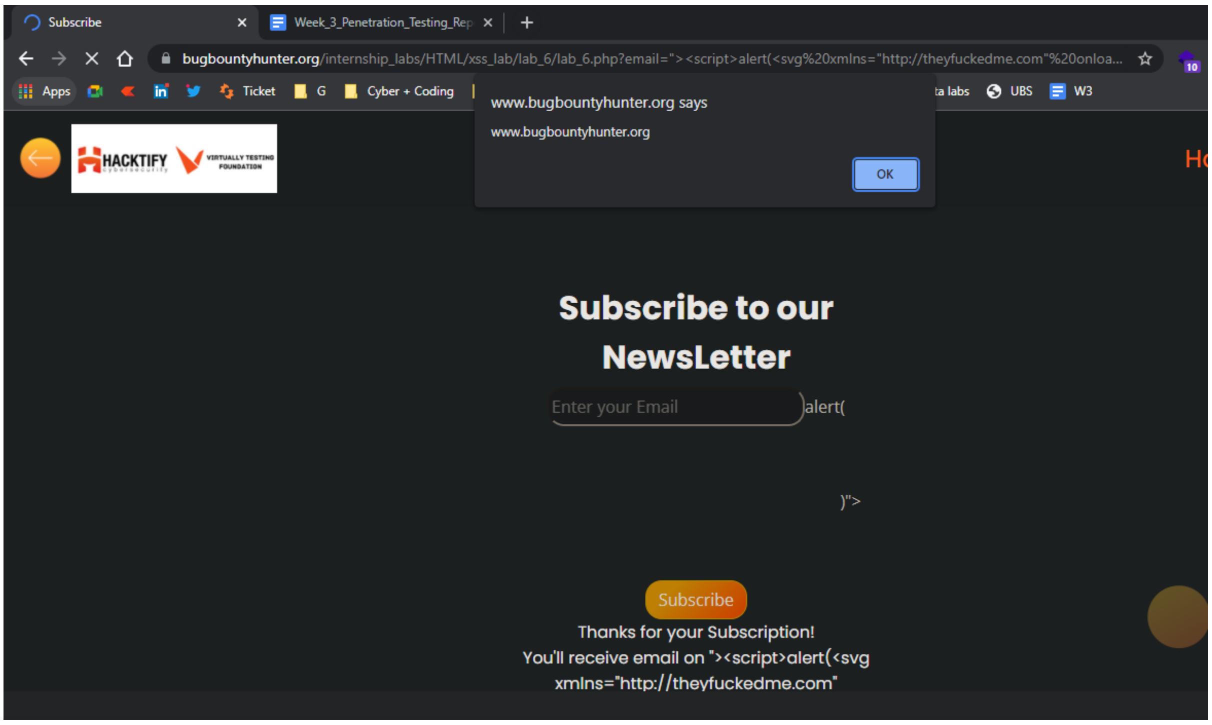
Subscribe

Thanks for your Subscription!
You'll receive email on "><img src=x onerror=alert(17)>

1.6. {Change the variation!}

Reference	**Risk Rating**
Change the variation!	**Hard**
Tools Used	
Google chrome	
Vulnerability Description	
I found this vulnerability by merging two different tags within a single command as payload to bypass the situation which the developer has created. I used, <script> tag in which I used <svg> tag to bypass it.	
How It Was Discovered	
Manual Analysis	
Vulnerable URLs	
https://www.bugbountyhunter.org/internship_labs/HTML/xss_lab/lab_6/lab_6.php?email=%22%3E%3Cscript%3Ealert(%3Csvg%20xmlns=%22http://theyfuckedme.com%22%20onload=%22alert(document.domain)%22/%3E)%3Cscript%3E	
Consequences of not Fixing the Issue	
Impersonate and masquerade as the victim user.	
Suggested Countermeasures	
Developers must sanitize the input of any user by denying entry of any tag or any special character if it is not a valid entry for that particular text field.	
References	
https://github.com/payloadbox/xss-payload-list	

Proof of Concept



bugbountyhunter.org/internship_labs/HTML/xss_lab/lab_6/lab_6.php?email="><script>alert(<svg%20xmlns="http://theyfuckedme.com"%20onloa...

www.bugbountyhunter.org says

www.bugbountyhunter.org

Subscribe to our NewsLetter

Enter your Email)alert(

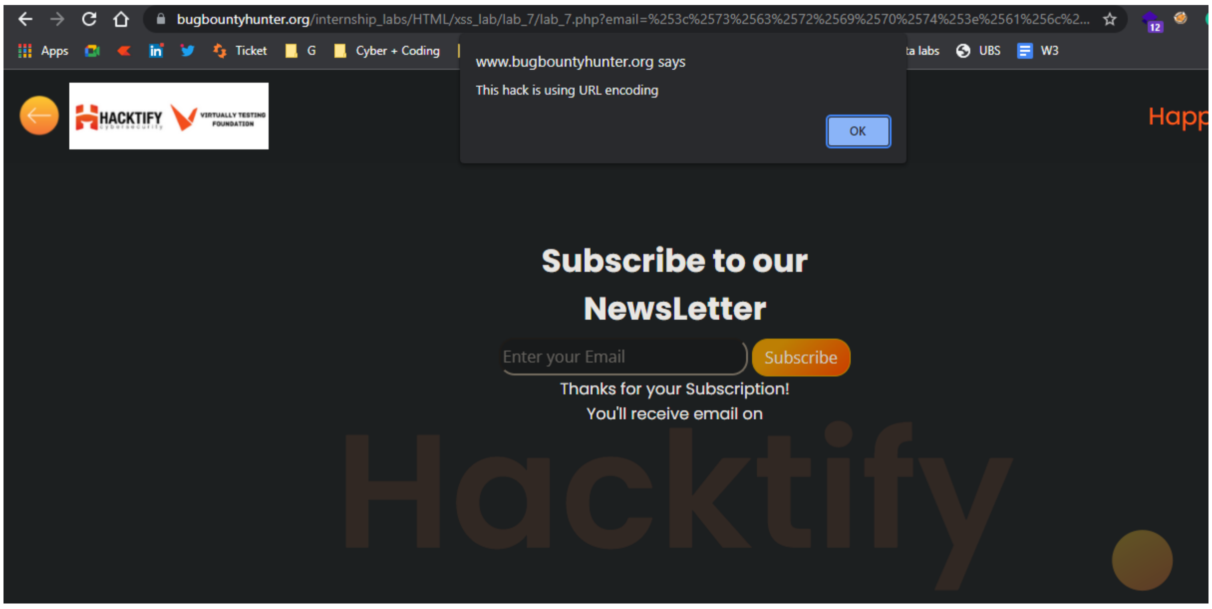
)">

Thanks for your Subscription!
You'll receive email on "><script>alert(<svg xmlns="http://theyfuckedme.com"

1.7. {Encoding is the key?}

Reference	**Risk Rating**
Encoding is the key?	**Medium**
Tools Used	
Google chrome and Burp Suite	
Vulnerability Description	
This vulnerability was found by URL encoding a simple javascript command with the help of Burp Suite.	
How It Was Discovered	
Manual Analysis and Automated Analysis	
Vulnerable URLs	
https://www.bugbountyhunter.org/internship_labs/HTML/xss_lab/lab_7/lab_7.php?email=%253c%2573%2563%2572%2569%2570%2574%253e%2561%256c%2565%2572%2574%2528%2522%2554%2568%2569%2573%2520%2568%2561%2563%256b%2520%2569%2573%2520%2575%2573%2569%256e%2567%2520%2555%2552%254c%2520%2565%256e%2563%256f%2564%2569%256e%2567%2522%2529%253c%252f%2573%2563%2572%2569%2570%2574%253e	
Consequences of not Fixing the Issue	
Attackers will be able to read the data that the user is able to access.	
Suggested Countermeasures	
Filtering inputs at the arrival of input data which will run through text fields.	
References	
https://github.com/payloadbox/xss-payload-list	

Proof of Concept


bugbountyhunter.org/internship_labs/HTML/xss_lab/lab_7/lab_7.php?email=%253c%2573%2563%2572%2569%2570%2574%253e%2561%256c%2...

www.bugbountyhunter.org says

This hack is using URL encoding

OK

Subscribe to our NewsLetter

Enter your Email

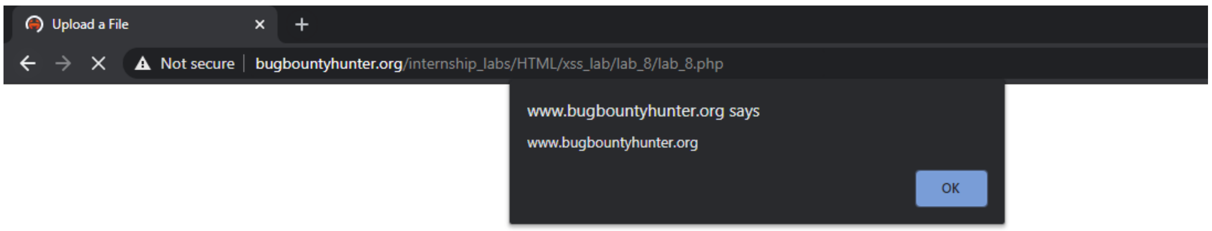
Subscribe

Thanks for your Subscription!
You'll receive email on

1.8. {XSS with file upload (file name)}

Reference	Risk Rating
XSS with file upload (file name)	**Low**
Tools Used	
Google chrome and Burp Suite	
Vulnerability Description	
I found this vulnerability by intercepting a file upload request onto Burp Suite tool and renaming the file name to insert payload into filename to bypass all the validation to load payload into the web pages and database to store payload into it.	
How It Was Discovered	
Manual Analysis and Automated Analysis	
Vulnerable URLs	
https://www.bugbountyhunter.org/internship_labs/HTML/xss_lab/lab_8/lab_8.php	
Consequences of not Fixing the Issue	
Hijacking of user accounts, stealing credentials, exfiltration of sensitive data	
Suggested Countermeasures	
Using appropriate HTTP headers and using Content Security Policy	
References	
https://github.com/payloadbox/xss-payload-list	

Proof of Concept


Upload a File
Not secure
bugbountyhunter.org/internship_labs/HTML/xss_lab/lab_8/lab_8.php
OK

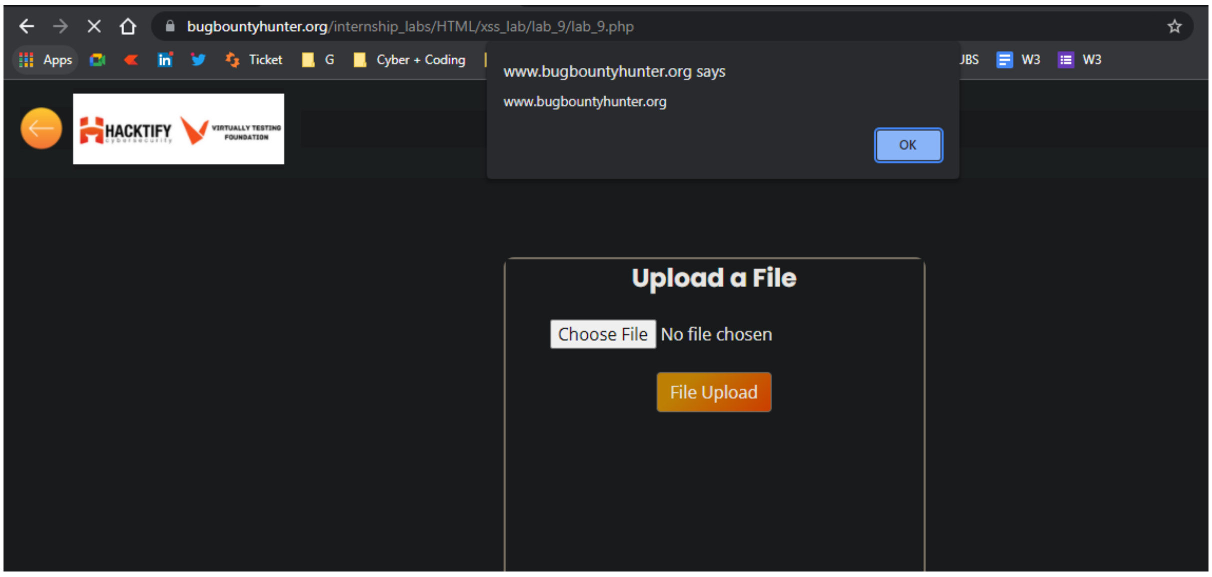
www.bugbountyhunter.org says

www.bugbountyhunter.org

1.9. {XSS with file upload (file content)}

Reference	Risk Rating
XSS with file upload (file content)	**Medium**
Tools Used	
Google chrome	
Vulnerability Description	
I found this vulnerability by uploading a malicious code file into the file upload section. In malicious code, I used <script> tag to design a malicious code.	
How It Was Discovered	
Manual Analysis	
Vulnerable URLs	
https://www.bugbountyhunter.org/internship_labs/HTML/xss_lab/lab_9/lab_9.php	
Consequences of not Fixing the Issue	
Hijacking of user accounts, stealing credentials, exfiltration of sensitive data	
Suggested Countermeasures	
Using appropriate HTTP headers and using Content Security Policy	
References	
https://github.com/payloadbox/xss-payload-list	

Proof of Concept


bugbountyhunter.org/internship_labs/HTML/xss_lab/lab_9/lab_9.php

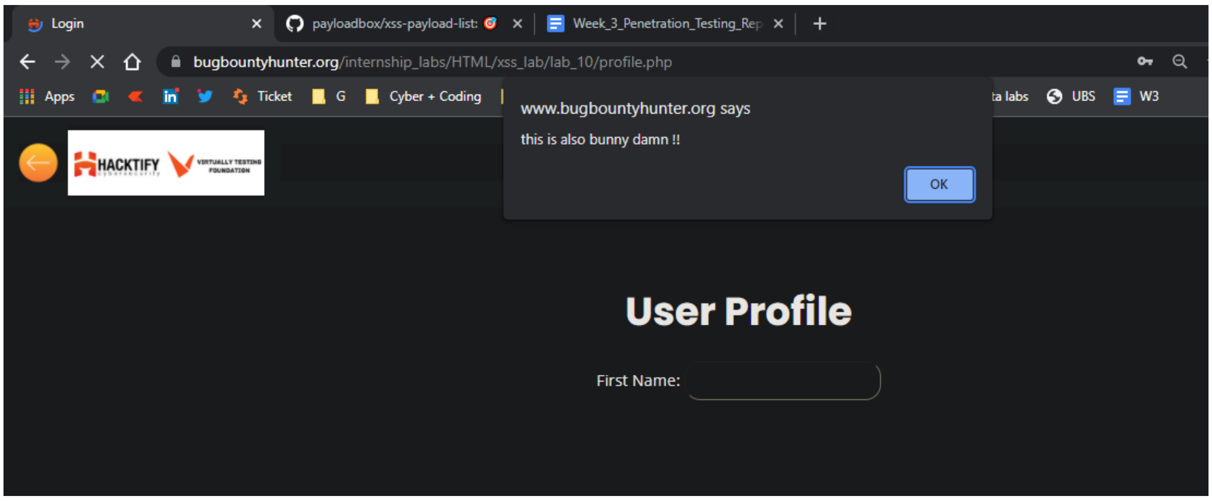
www.bugbountyhunter.org says

www.bugbountyhunter.org

1.10. {Stored Everywhere}

Reference	**Risk Rating**
Stored Everywhere	**Low**
Tools Used	
Google chrome	
Vulnerability Description	
I found this vulnerability by registering my into register web page. I inserted a simple <script> tag in all fields which are required to fill for registration. After registration, I tried logging in into the webpage and after the login request I was able to verify that I have successfully implemented a stored XSS attack. Here, these credentials will be stored into the database which can affect other used credentials as well as other database related features.	
How It Was Discovered	
Manual Analysis	
Vulnerable URLs	
https://www.bugbountyhunter.org/internship_labs/HTML/xss_lab/lab_10/profile.php	
Consequences of not Fixing the Issue	
Hijacking of user accounts, stealing credentials, exfiltration of sensitive data	
Suggested Countermeasures	
Using appropriate HTTP headers and using Content Security Policy	
References	
https://github.com/payloadbox/xss-payload-list	

Proof of Concept


Login
payloadbox/xss-payload-list:
Week_3_Penetration_Testing_Rep
bugbountyhunter.org/internship_labs/HTML/xss_lab/lab_10/profile.php
Apps
Ticket
G
Cyber + Coding
a labs
UBS
W3
HACKTIFY
OK
User Profile
First Name:

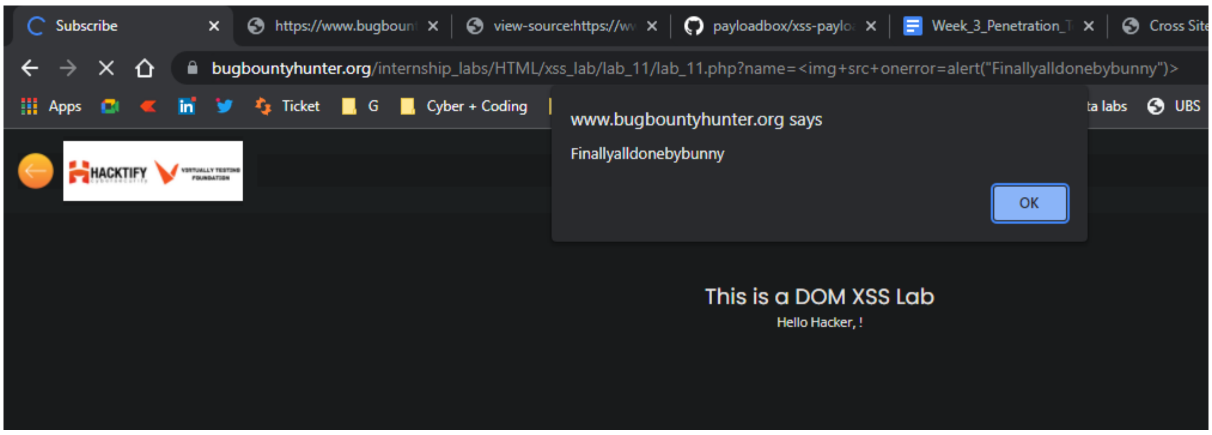
www.bugbountyhunter.org says

this is also bunny damn !!

1.11. {DOM's are Love}

Reference	**Risk Rating**
DOM's are Love	**Hard**
Tools Used	
Google chrome	
Vulnerability Description	
I found this vulnerability by inspecting the source code of the web page and finding out what variables are used in this section. And later on tried to load payload using <img> tag.	
How It Was Discovered	
Manual Analysis	
Vulnerable URLs	
https://www.bugbountyhunter.org/internship_labs/HTML/xss_lab/lab_11/lab_11.php?name=%3Cimg+src+onerror=alert(%22Finallyalldonebybunny%22)%3E	
Consequences of not Fixing the Issue	
Hijacking of user accounts, stealing credentials, exfiltration of sensitive data	
Suggested Countermeasures	
Using appropriate HTTP headers and using Content Security Policy	
References	
https://github.com/payloadbox/xss-payload-list	

Proof of Concept



bugbountyhunter.org/internship_labs/HTML/xss_lab/lab_11/lab_11.php?name=<img+src+onerror=alert("Finallyalldonebybunny")>

www.bugbountyhunter.org says

Finallyalldonebybunny

This is a DOM XSS Lab

Hello Hacker, !