**Week 03**

**Penetration Testing Report**

**Introduction**

This report document hereby describes the proceedings and results of a HTML injection & Clickjackings assessment conducted against the **Week 03 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 03 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** | **Lab 1 -** [Cross Site Scripting](https://www.bugbountyhunter.org/internship_labs/HTML/xss_lab/index.php) |
| --- | --- |

**3. Summary**

Outlined is a Black Box Application Security assessment for the **Week 03 Labs**.

**Total number of Sub-labs: 8 Sub-labs**

| **High** | **Medium** | **Low** |
| --- | --- | --- |
| **3** | **3** | **05** |

**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. Lab 1 [Cross Site Scripting](https://www.bugbountyhunter.org/internship_labs/HTML/xss_lab/index.php)

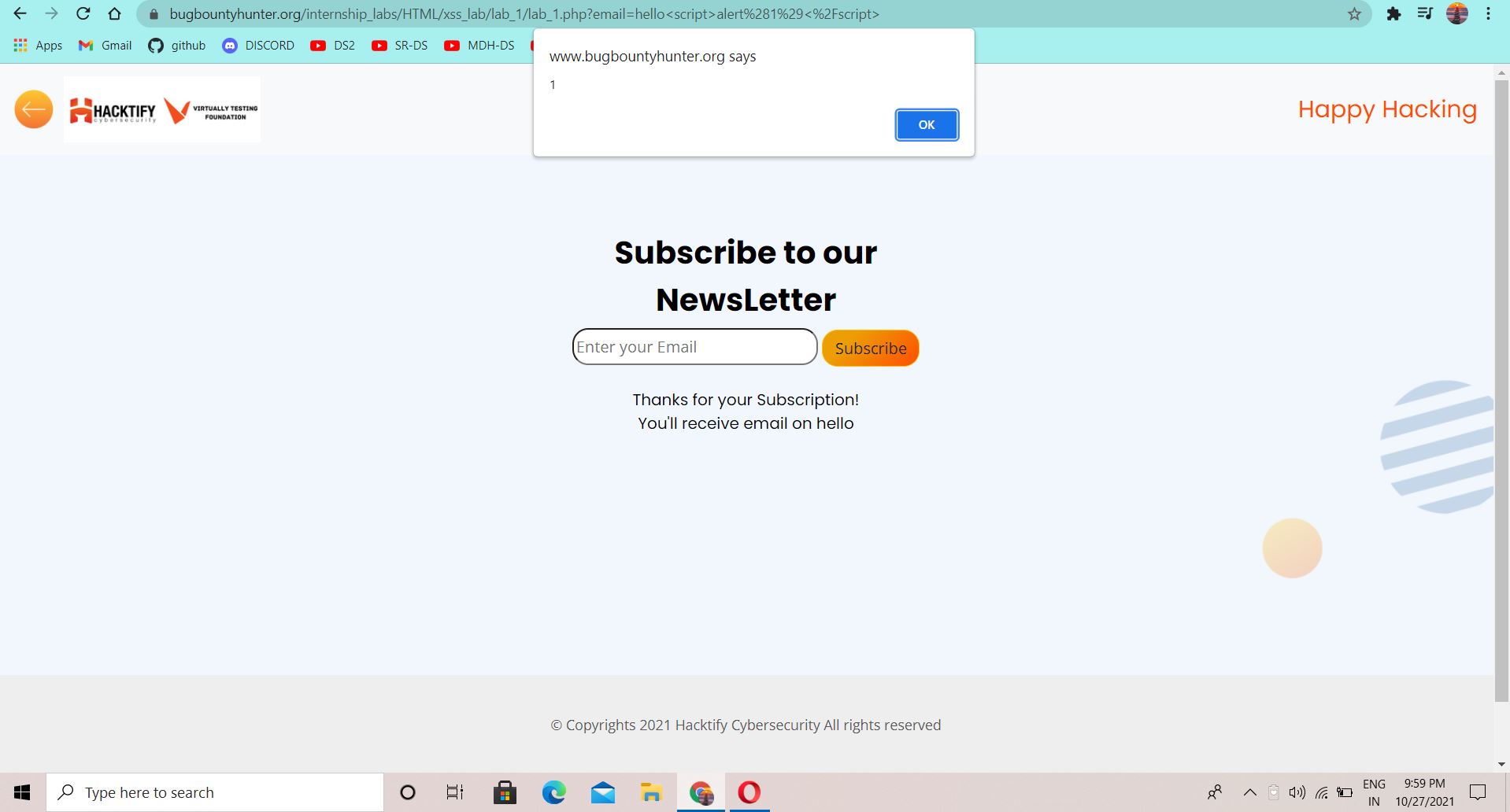
# 

# 1.1. [Let's Do IT!](https://www.bugbountyhunter.org/internship_labs/HTML/xss_lab/lab_1/index.php)

| **Reference** | **Risk Rating** |
| --- | --- |
| Sub-lab-1 Let’s Do IT! | **Low** |
| **Tools Used** | |
| Tools that you have used to find the vulnerability. : Web browser | |
| **Vulnerability Description** | |
| Cross-site scripting (also known as XSS) is a web security vulnerability that allows an attacker to compromise the interactions that users have with a vulnerable application. It allows an attacker to circumvent the same origin policy, which is designed to segregate different websites from each other. Cross-site scripting vulnerabilities normally allow an attacker to masquerade as a victim user, to carry out any actions that the user is able to perform, and to access any of the user's data. If the victim user has privileged access within the application, then the attacker might be able to gain full control over all of the application's functionality and data. It provides how XSS works in real-life scenario and how one should have a mindset when hunting for XSS vulnerabilities. The XSS we saw in this lab was Reflected XSS which has a severity of P3 with a CVSS score of 5.8 which is Medium. | |
| **How It Was Discovered** | |
| Manual Analysis | |
| **Vulnerable URLs** | |
|  | |
| **Consequences of not Fixing the Issue** | |
| Attackers will often attempt to exploit unpatched flaws or access default accounts, unused pages,malicious links and gain unauthorized access or knowledge of the system. | |
| **Suggested Countermeasures** | |
| Restrict their access from injecting different xss attacks | |
| **References** | |
|  | |

# 

# Proof of Concept



**1.2.**[**Balancing Is Important In Life!**](https://www.bugbountyhunter.org/internship_labs/HTML/xss_lab/lab_2/index.php)

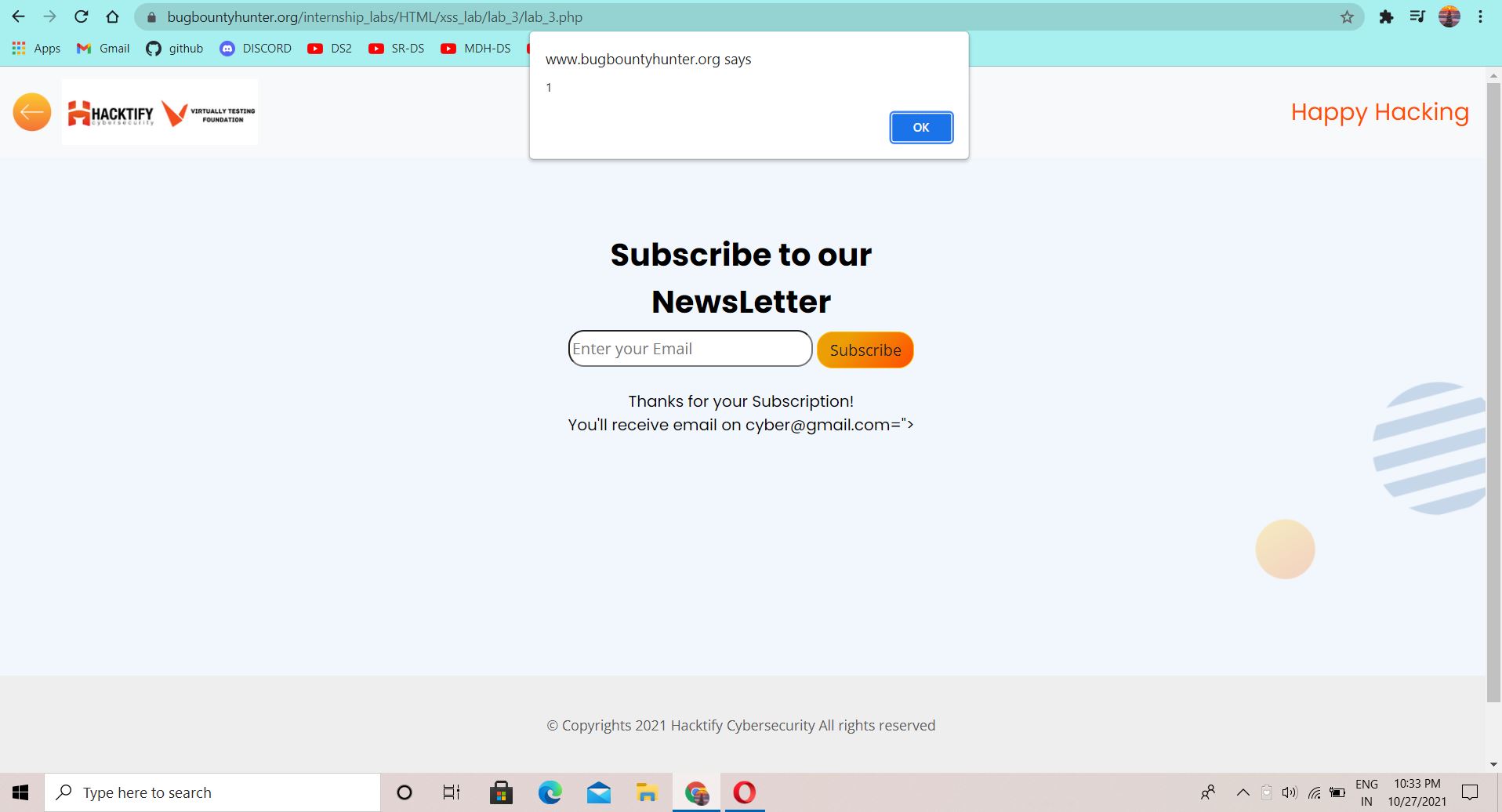
| **Reference** | **Risk Rating** |
| --- | --- |
| Sub-lab-2 Balancing is important in life! | **Low** |
| **Tools Used** | |
| Browser | |
| **Vulnerability Description** | |
| The attacker can insert unauthorized JavaScript, VBScript, HTML, or other active content into a web page viewed by other users. A malicious script inserted into a page in this manner can hijack the user’s session, submit unauthorized transactions as the user, steal confidential information, or simply deface the page. attempt to provide how one can perform XSS attack by simply balancing the payload. The XSS we saw in this lab was Reflected XSS which has a severity of P3 with a CVSS score of 5.8 which is Medium. | |
| **How It Was Discovered** | |
| Manual Analyis | |
| **Vulnerable URLs** | |
|  | |
| **Consequences of not Fixing the Issue** | |
|  | |
| **Suggested Countermeasures** | |
| implementing a validation mechanism for all the data received from the user | |
| **References** | |
|  | |

# Proof of Concept

**1.3.** [**XSS Is Everywhere**](https://www.bugbountyhunter.org/internship_labs/HTML/xss_lab/lab_3/index.php)**!**

| **Reference** | **Risk Rating** |
| --- | --- |
| Sub-lab-3 XSS Is Everywhere! | **Low** |
| **Tools Used** | |
| Browser | |
| **Vulnerability Description** | |
| This was an attempt to provide how one can perform XSS attack in email address fields. The XSS we saw in this lab was Reflected XSS which has a severity of P3 with a CVSS score of 5.8 which is Medium | |
| **How It Was Discovered** | |
| Manual Analysis | |
| **Vulnerable URLs** | |
|  | |
| **Consequences of not Fixing the Issue** | |
|  | |
| **Suggested Countermeasures** | |
| XSS prevention should not be about filtering, but proper encoding. All dangerous characters (in most contexts - [but not all](https://www.owasp.org/index.php/XSS_%28Cross_Site_Scripting%29_Prevention_Cheat_Sheet#XSS_Prevention_Rules) - these are at least ', ", <, and >) should be HTML encoded when send to a user, no matter what filter was applied beforehand. | |
| **References** | |
| High-Tech Bridge Advisory HTB23241 - https://www.immuniweb.com/advisory/HTB23241 - Reflected Cross-Site Scripting (XSS) in Simple Email Form Joomla Extension. | |

# Proof of Concept



# 

# 

# 1.4. [Alternatives Are Must!](https://www.bugbountyhunter.org/internship_labs/HTML/xss_lab/lab_4/index.php)

| **Reference** | **Risk Rating** |
| --- | --- |
| Sub-lab-4 Alternatives are Must! | **Medium** |
| **Tools Used** | |
| Browser | |
| **Vulnerability Description** | |
| This was an attempt to provide how one can perform XSS when alert is blocked . In such a case the alternatives confirm or prompt can work. The XSS we saw in this lab was Reflected XSS which has a severity of P3 with a CVSS score of 5.8 which is Medium. | |
| **How It Was Discovered** | |
| Manual Analysis | |
| **Vulnerable URLs** | |
|  | |
| **Consequences of not Fixing the Issue** | |
|  | |
| **Suggested Countermeasures** | |
|  | |
| **References** | |
|  | |

# Proof of Concept

# 1.5. [Developer Hates Scripts!](https://www.bugbountyhunter.org/internship_labs/HTML/xss_lab/lab_5/index.php)

| **Reference** | **Risk Rating** |
| --- | --- |
| Sub-lab-5 Developer Hates Scripts! | **Hard** |
| **Tools Used** | |
|  | |
| **Vulnerability Description** | |
| This was an attempt to provide how one can perform XSS when <script> tags are blocked . In such a case other payloads like <img src=x onerror=alert(1)> can be very useful. The XSS we saw in this lab was Reflected XSS which has a severity of P3 with a CVSS score of 5.8 which is Medium. | |
| **How It Was Discovered** | |
| Manual Analysis | |
| **Vulnerable URLs** | |
|  | |
| **Consequences of not Fixing the Issue** | |
|  | |
| **Suggested Countermeasures** | |
|  | |
| **References** | |
|  | |

# 

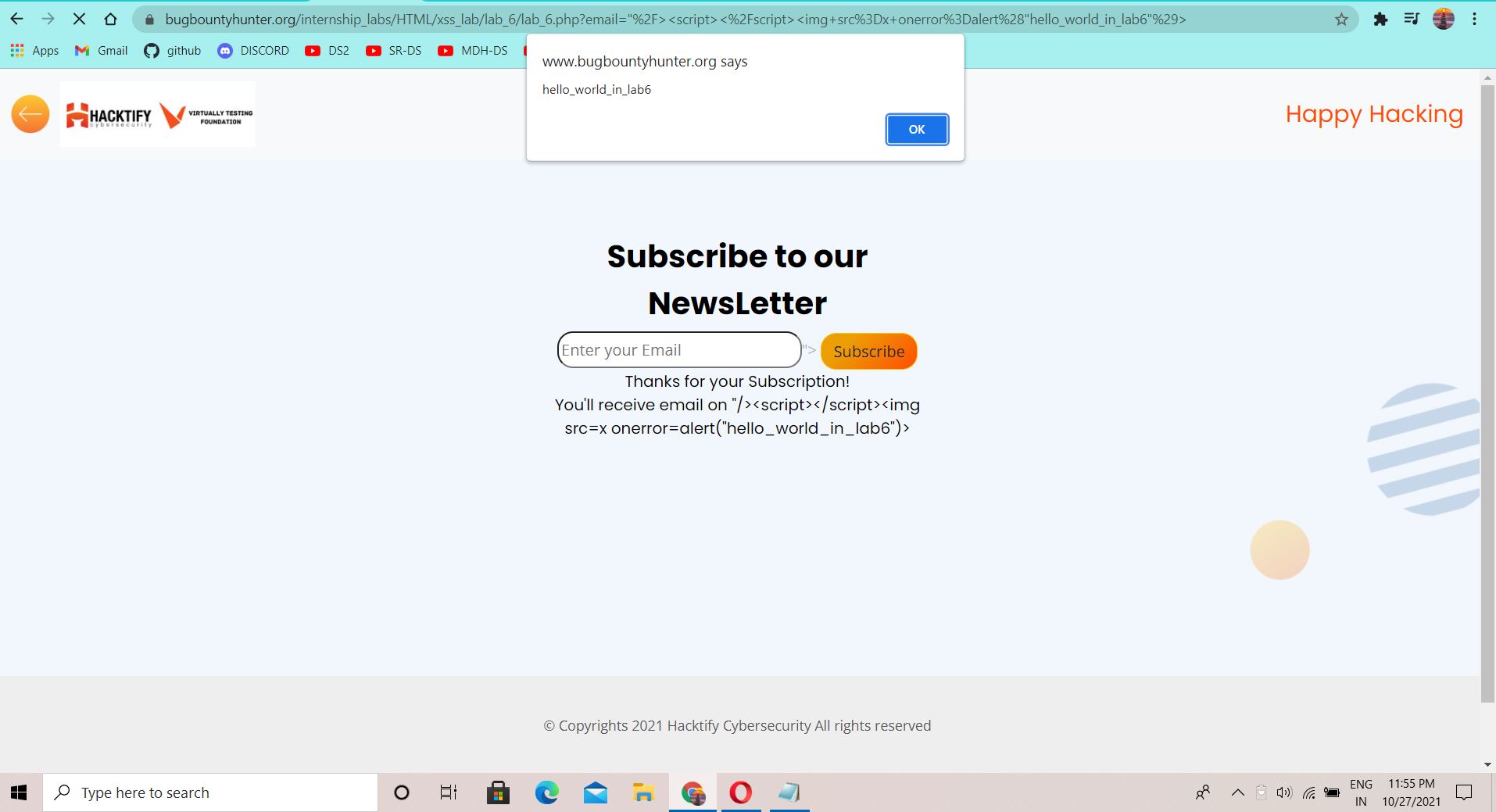
# Proof of Concept

# 

# 1.6. [Change The Variation!](https://www.bugbountyhunter.org/internship_labs/HTML/xss_lab/lab_6/index.php)

| **Reference** | **Risk Rating** |
| --- | --- |
| Sub-lab-6 Change The Variation! | **High** |
| **Tools Used** | |
| Web Browser | |
| **Vulnerability Description** | |
| This was an attempt to provide how one can perform XSS when <script> tags are blocked and never reflected directly. In such a case other payloads like "><scr<script>ipt>alert(1)</scr</script>ipt> can be very useful. The XSS we saw in this lab was Reflected XSS which has a severity of P3 with a CVSS score of 5.8 which is Medium. | |
| **How It Was Discovered** | |
| Manual Analysis | |
| **Vulnerable URLs** | |
|  | |
| **Consequences of not Fixing the Issue** | |
|  | |
| **Suggested Countermeasures** | |
|  | |
| **References** | |
|  | |

# Proof of Concept



# 1.7. [Encoding Is The Key](https://www.bugbountyhunter.org/internship_labs/HTML/xss_lab/lab_7/index.php)?

| **Reference** | **Risk Rating** |
| --- | --- |
| Sub-lab-7 Encoding Is The Key? | **Medium** |
| **Tools Used** | |
| Web Browser | |
| **Vulnerability Description** | |
| The attack aims to explore flaws in the decoding mechanism implemented on applications when decoding Unicode data format. An attacker can use this technique to encode certain characters in the URL to bypass application filters, thus accessing restricted resources on the Web server or to force browsing to protected pages. | |
| **How It Was Discovered** | |
| Manual Analysis | |
| **Vulnerable URLs** | |
|  | |
| **Consequences of not Fixing the Issue** | |
| The communications between components can be modified in unexpected ways. Unexpected commands can be executed, bypassing other security mechanisms. Incoming data can be misinterpreted. | |
| **Suggested Countermeasures** | |
| Never Insert Untrusted Data Except in Allowed Locations | |
| **References** | |
|  | |

# Proof of Concept

# 

# 

# 

# 

# 

# 

# 

# 

# 1.8. [XSS With File Upload (File Name)](https://www.bugbountyhunter.org/internship_labs/HTML/xss_lab/lab_8/index.php)[!](https://www.bugbountyhunter.org/internship_labs/HTML/xss_lab/lab_6/index.php)

| **Reference** | **Risk Rating** |
| --- | --- |
| Sub-lab-8 XSS With File Upload(File Name)! | **Low** |
| **Tools Used** | |
| Web Browser | |
| **Vulnerability Description** | |
|  | |
| **How It Was Discovered** | |
| Manual Analysis | |
| **Vulnerable URLs** | |
|  | |
| **Consequences of not Fixing the Issue** | |
|  | |
| **Suggested Countermeasures** | |
|  | |
| **References** | |
|  | |

# Proof of Concept

# 

# 

# 1.9. [XSS With File Upload (File Content)](https://www.bugbountyhunter.org/internship_labs/HTML/xss_lab/lab_9/index.php)!

# 

| **Reference** | **Risk Rating** |
| --- | --- |
| Sub-lab-9 XSS With File Upload(File Content! | **Medium** |
| **Tools Used** | |
| Web Browser | |
| **Vulnerability Description** | |
|  | |
| **How It Was Discovered** | |
| Manual Analysis | |
| **Vulnerable URLs** | |
|  | |
| **Consequences of not Fixing the Issue** | |
|  | |
| **Suggested Countermeasures** | |
|  | |
| **References** | |
|  | |

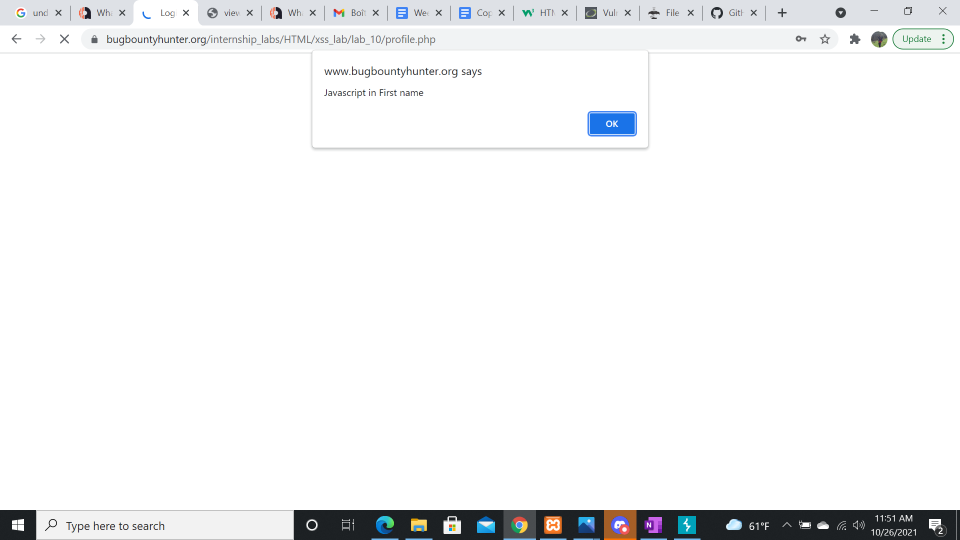
# Proof of Concept

# 

# 1.10. [Stored Everywhere](https://www.bugbountyhunter.org/internship_labs/HTML/xss_lab/lab_10/index.php)!

| **Reference** | **Risk Rating** |
| --- | --- |
| Sub-lab-10 Stored Everywhere! | **Low** |
| **Tools Used** | |
| Web Browser | |
| **Vulnerability Description** | |
| it provide how one can perform Stored XSS when our input gets stored in a database. The XSS we saw in this lab was Stored XSS which has a severity of P2 with a CVSS score of 7-8.9 which is High. | |
| **How It Was Discovered** | |
| Manual Analysis | |
| **Vulnerable URLs** | |
|  | |
| **Consequences of not Fixing the Issue** | |
|  | |
| **Suggested Countermeasures** | |
|  | |
| **References** | |
|  | |

# Proof of Concept



# 

# 

# 1.11. [DOM's Are Love!](https://www.bugbountyhunter.org/internship_labs/HTML/xss_lab/lab_11/index.php)

| **Reference** | **Risk Rating** |
| --- | --- |
| Sub-lab-11 DOM’s Are Love! | **High** |
| **Tools Used** | |
| Web Browser | |
| **Vulnerability Description** | |
|  | |
| **How It Was Discovered** | |
| Manual Analysis | |
| **Vulnerable URLs** | |
|  | |
| **Consequences of not Fixing the Issue** | |
|  | |
| **Suggested Countermeasures** | |
|  | |
| **References** | |
|  | |

# Proof of Concept