**Penetration Testing Report**

**Full Name:Ilyas Bajji  
Program: HCPT**

**Date:20/03/2024**

**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** | **Server-Side Request Forgery, Cross-Site Request Forgery** |

**3. Summary**

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

**Total number of Sub-labs: {count} Sub-labs**

|  |  |  |
| --- | --- | --- |
| **High** | **Medium** | **Low** |
| **4** | **5** | **6** |

**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. Server-Side Request Forgery

# 1.1. Get the 127.0.0.1

|  |  |
| --- | --- |
| **Reference** | **Risk Rating** |
| Get the 127.0.0.1 | **Low** |
| **Tools Used** | |
| observation | |
| **Vulnerability Description** | |
| Manipulate server-side requests made to internal or external resources | |
| **How It Was Discovered** | |
| Inject in url parameter : 127.0.0.1:80 | |
| **Vulnerable URLs** | |
| https://labs.hacktify.in/HTML/ssrf\_lab/lab\_1/lab\_1.php | |
| **Consequences of not Fixing the Issue** | |
| URL-based SSRF, Protocol-based SSRF, Blind SSRF | |
| **Suggested Countermeasures** | |
| Input Validation, Whitelisting, Network Segmentation, CSP (Content Security Policy) | |
| **References** | |
| https://portswigger.net/web-security/ssrf | |

# 

# Proof of Concept

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

A screenshot of a computer

Description automatically generated

# 1.2. Http(S)? Nevermind !!

|  |  |
| --- | --- |
| **Reference** | **Risk Rating** |
| Http(S)? Nevermind !! | **Low** |
| **Tools Used** | |
| observation | |
| **Vulnerability Description** | |
| Manipulate server-side requests made to internal or external resources | |
| **How It Was Discovered** | |
| Inject in url parameter : http://localhost:80 | |
| **Vulnerable URLs** | |
| URLs of the vulnerable pages in the lab | |
| **Consequences of not Fixing the Issue** | |
| URL-based SSRF, Protocol-based SSRF, Blind SSRF | |
| **Suggested Countermeasures** | |
| Input Validation, Whitelisting, Network Segmentation, CSP (Content Security Policy) | |
| **References** | |
| https://portswigger.net/web-security/ssrf | |

# Proof of Concept

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

A screenshot of a computer

Description automatically generated