Deep Kakadiya

Title: Reflected XSS in Search Field

Date : 1/08/2024

Introduction : found a "Reflected XSS " in the search field in the website **http://testasp.vulnweb.com** as part of my project tasks at INTERNSHIP STUDIO. This report details the vulnerability I discovered, the steps I took to exploit it, and provides recommendations for remediation.

Vulnerability Details:

• **Vulnerability Type:** Reflected XSS

· **Affected Input Field:** The vulnerable input field is the search query parameter located in the URL.

· **Vulnerable URL:**
http://testasp.vulnweb.com/Search.as

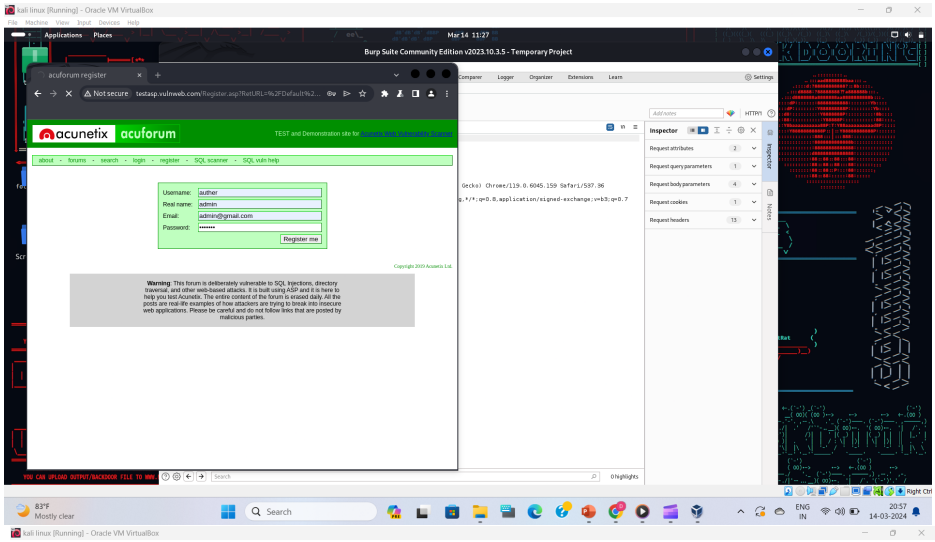
· **Payload:** I crafted a simple payload to demonstrate the vulnerability.

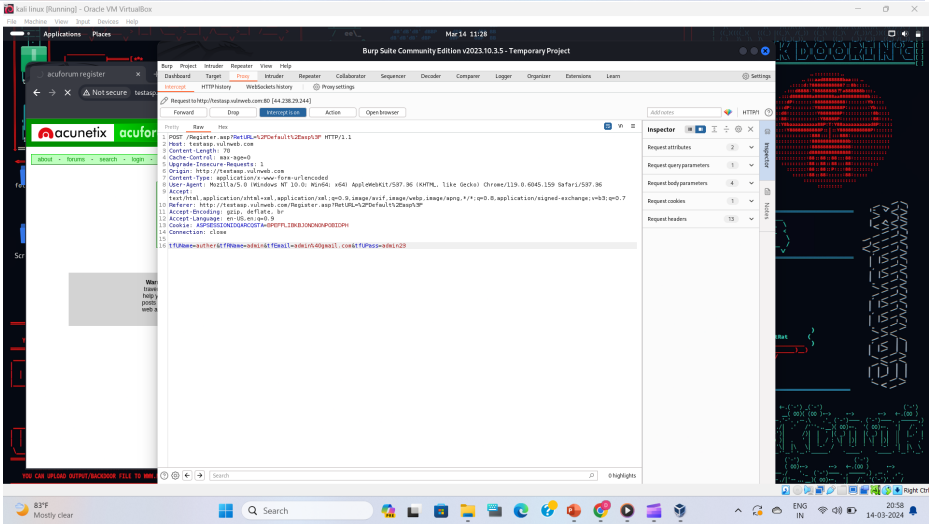
<script>alert(1)</script>

Exploitation:

When the payload is inserted into the search query parameter and the page is loaded, the JavaScript code is executed in the user's browser, triggering the alert.

Screenshots :





Impact: An attacker could exploit this vulnerability to execute arbitrary JavaScript code within the context of the victim's browser.

This could lead to unauthorized access to user data, session hijacking, or phishing attacks.

Proof of Concept (PoC): To replicate the issue, follow these steps:

1. Visit the vulnerable URL:

http://testasp.vulnweb.com/Search.asp

2. Select the search section

3. Inset the payload in the search box .

4. The alert should pop up, indicating successful execution of the XSS payload.

Recommendations:

To mitigate this vulnerability, the following steps should be taken

1.Input Sanitization: Properly sanitize and validate user inputs before they are reflected in HTML content. Encode any user-provided data to prevent execution of malicious scripts.

2. Content Security Policy (CSP): Implement a CSP to restrict the sources from which scripts can be executed, reducing the risk of successful XSS attacks.

3. Output Encoding: Always encode user-generated content that is displayed within HTML context to prevent script execution.