**Cross Site Scripting (XSS) in Vulnweb**

**Date of Discovery:** July 24, 2024

**Vulnerable Domain:** testasp.vulnweb.com (subdomain of Vulnweb)

**Severity:** High

**Vulnerability Type:** Cross-Site Scripting (XSS)

**Description:**

A Cross-Site Scripting (XSS) vulnerability has been discovered on the testasp.vulnweb.com subdomain. The issue arises when the web application fails to properly sanitize user input, allowing the injection of malicious JavaScript code.

**Steps to reproduce: -**

1. Go to  <http://testasp.vulnweb.com/> and click on the Search button.
2. Then in the search box enter the following code “<script>alert(1)</script>”
3. After entering the code, click on search and it will give you a popup of the alert.
4. That way you can see a vulnerability of XSS.

**Proof of Concept (PoC):**

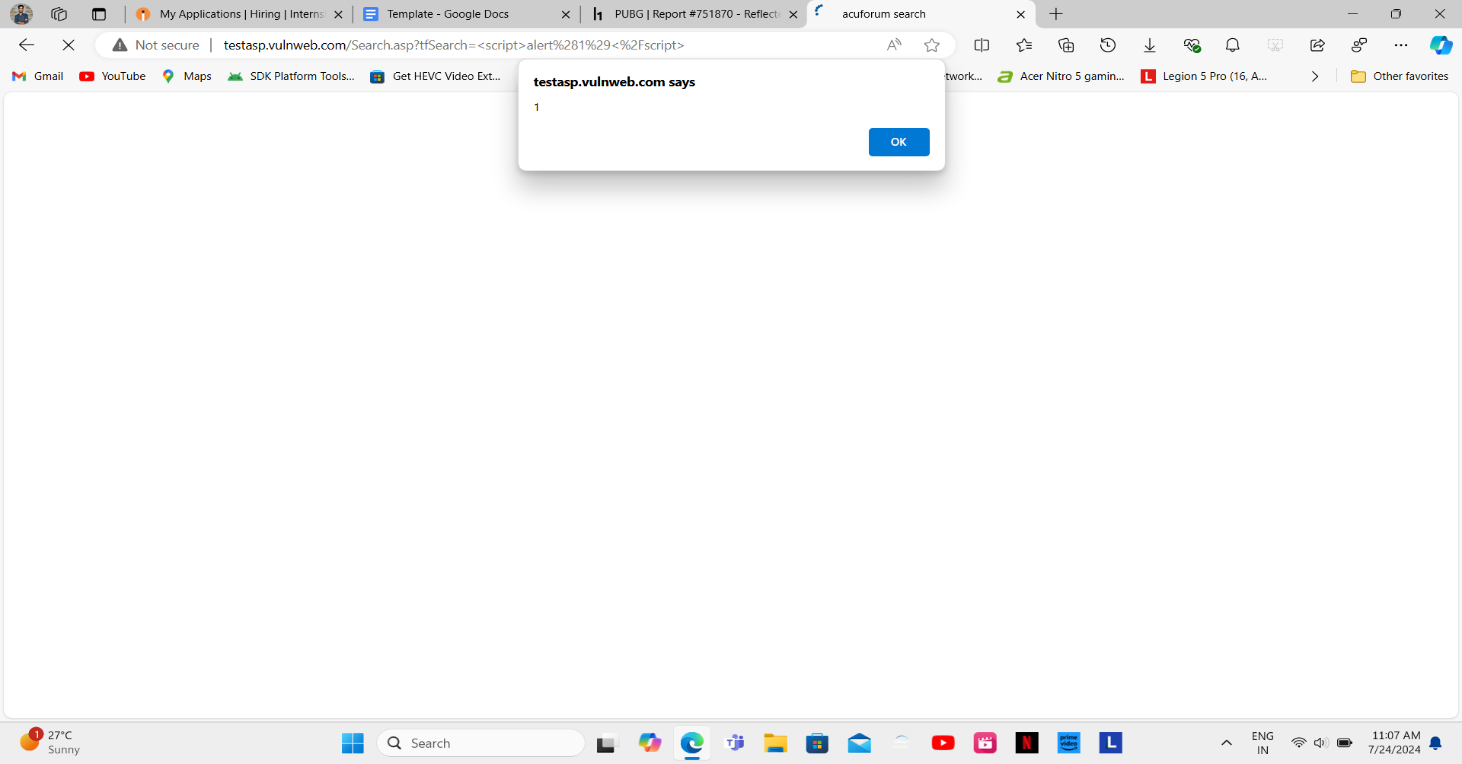
The following input, when submitted through a form or included in a URL parameter, triggers the XSS vulnerability:

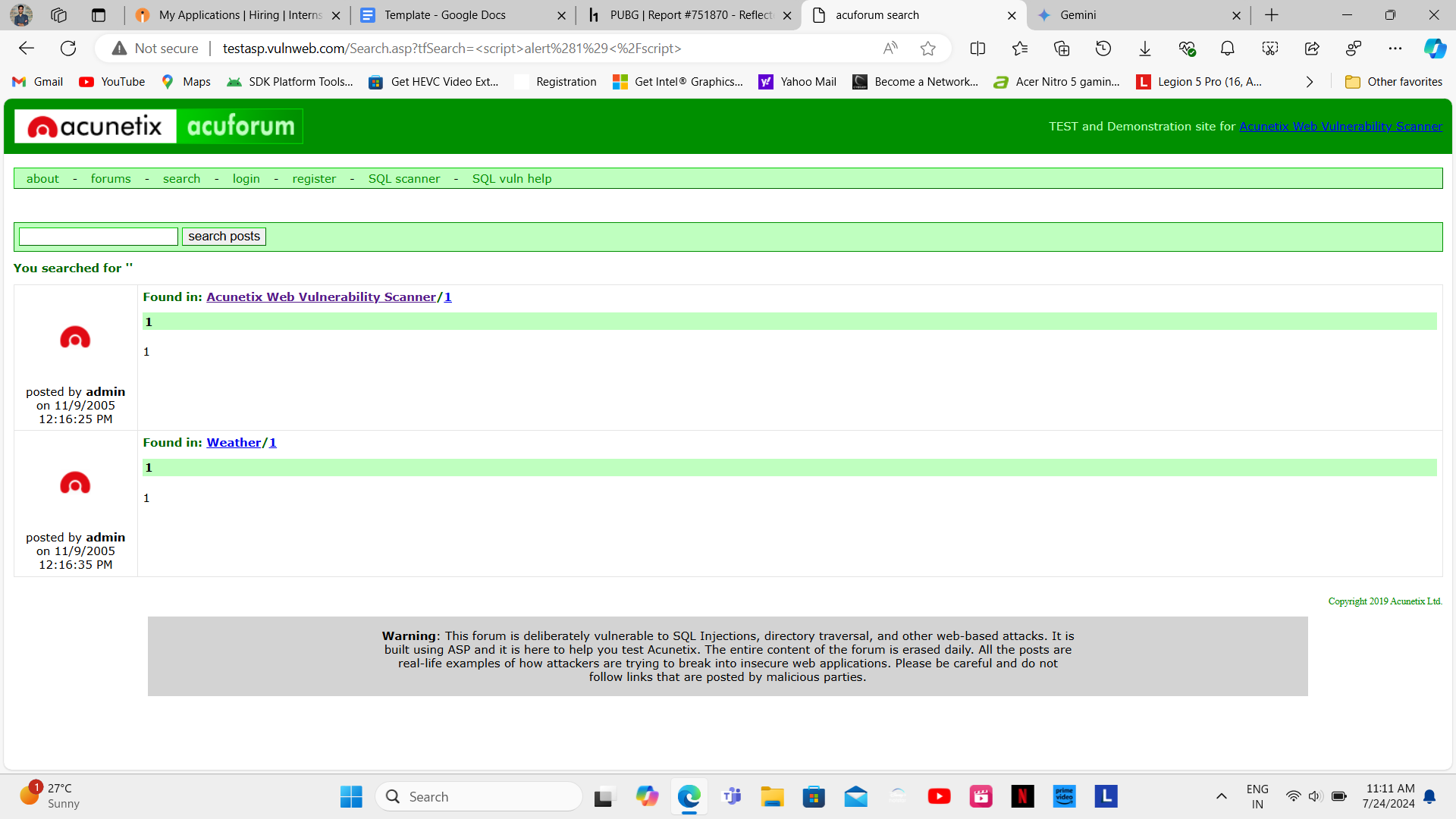
HTML

<script>alert(1)</script>

This results in the execution of the alert(1) JavaScript function, which pops up a dialog box displaying the number '1'.

**Request PoC: -**

****

****

**Video Demonstration: -**

****

**Impact:**

This XSS vulnerability poses a significant security risk. An attacker could exploit it to:

* **Steal User Sessions:** By injecting JavaScript code to capture cookies or authentication tokens, an attacker can hijack user sessions.
* **Perform Actions on Behalf of the User:** The injected script can manipulate the web page, submit forms, or perform other actions, potentially tricking the user into disclosing sensitive information.
* **Deface the Website:** An attacker can modify the website's content, potentially causing damage to reputation or disseminating false information.
* **Spread Malware:** By redirecting the user to malicious websites or injecting scripts to download and execute malware, an attacker can compromise the user's system.

**Mitigation:**

To mitigate this vulnerability, the following actions are recommended:

1. **Input Sanitization:** Implement robust input validation and sanitization on all user-provided data.
2. **Output Encoding:** Properly encode all data before rendering it on the web page.
3. **Content Security Policy (CSP):** Consider implementing a CSP to further restrict the execution of scripts.
4. **Regular Security Reviews:** Conduct periodic security reviews to identify and address potential vulnerabilities before they can be exploited.