Title: Cross Site Scripting

Domain: Velnweb.com

Subdomain: testasp.velnweb.com

Summary:

http://testasp.vulnweb.com/ has a main website Acunetix Web Vulnerability Scanner that has a Reflected Cross-Site Scripting (XSS) into HTML context that refers to a type of security vulnerability where an attacker injects malicious scripts into web pages that are dynamically generated by the server. These scripts are then executed in the context of the victim's browser when they access the web page. The "reflected" part of Reflected XSS means the payload is reflected immediately back to the user in the response, without being stored by the web server.

Steps To Reproduce:

Step 1: Visit the http://testasp.vulnweb.com/ Website

Step 2: On the top menu bar you will find the search option

Step 3: As it contains a simple reflected cross-site scripting vulnerability in the search functionality. I had to perform a cross-site scripting attack that calls the alert function.

Step 4: To perform a Cross-site Scripting in the search option type “<script>alert(1)</script>” and click on search post and solve the vulnerability.

Impact: Unlike stored XSS, where the payload is saved on the server (in a database or file), reflected XSS relies on data being sent as part of a request (often in the URL or form submission) and then reflected back by the server in the response.

- The core impact of reflected XSS is the execution of arbitrary JavaScript on the victim's browser.
- One of the most common impacts of XSS is the ability to steal authentication cookies or tokens.
- XSS allows attackers to inject fake content, such as fake login forms, onto a page.

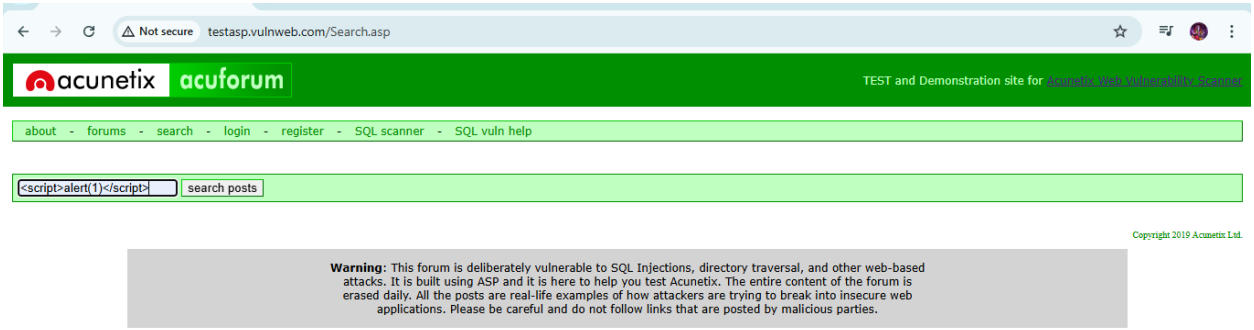
Mitigation:

Reflected XSS allows attackers to run harmful scripts on users' browsers, leading to data theft, phishing, and reputation damage. Preventing it requires careful input validation, output encoding, and the use of security policies like CSP.

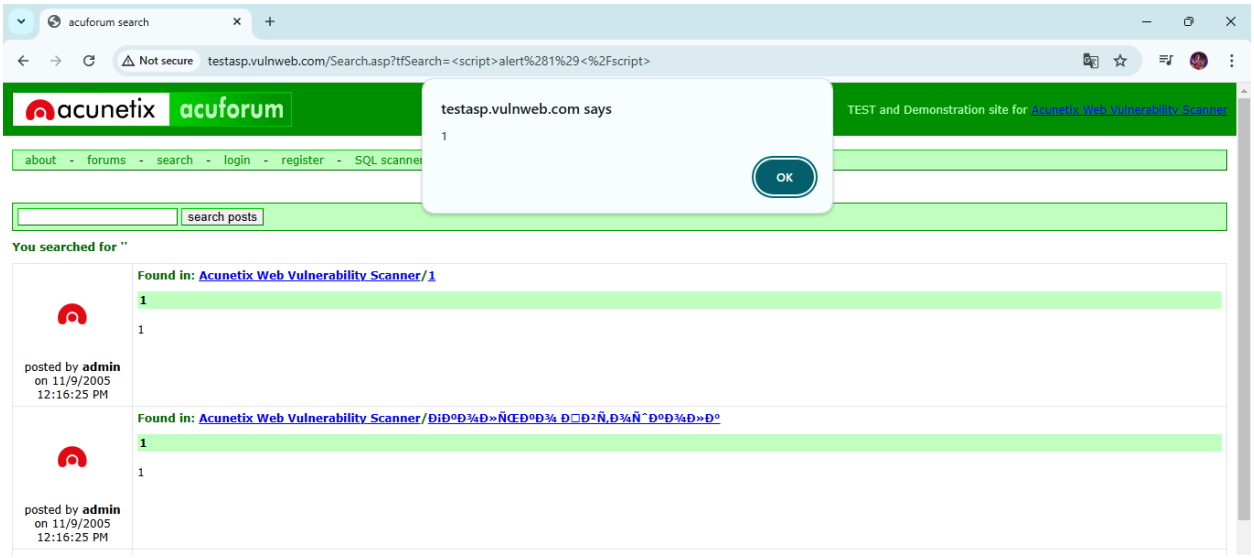
- **Sanitize Input**: Validate and clean user inputs to prevent script injection.
- **Output Encoding**: Encode dynamic content properly before inserting it into HTML.
- **Content Security Policy (CSP)**: Restrict what content can be executed on the page
- **Use Secure Cookies**: Mark sensitive cookies as `HttpOnly` and `Secure` to prevent access via JavaScript.

Request PoC including Screeshots is included in the report which is attached below:

- Image 1.png



- Image 2.png



- Screen Recording

Recording 2024-11-21 125219.mp4