

# Cross Site Scripting in vulnweb

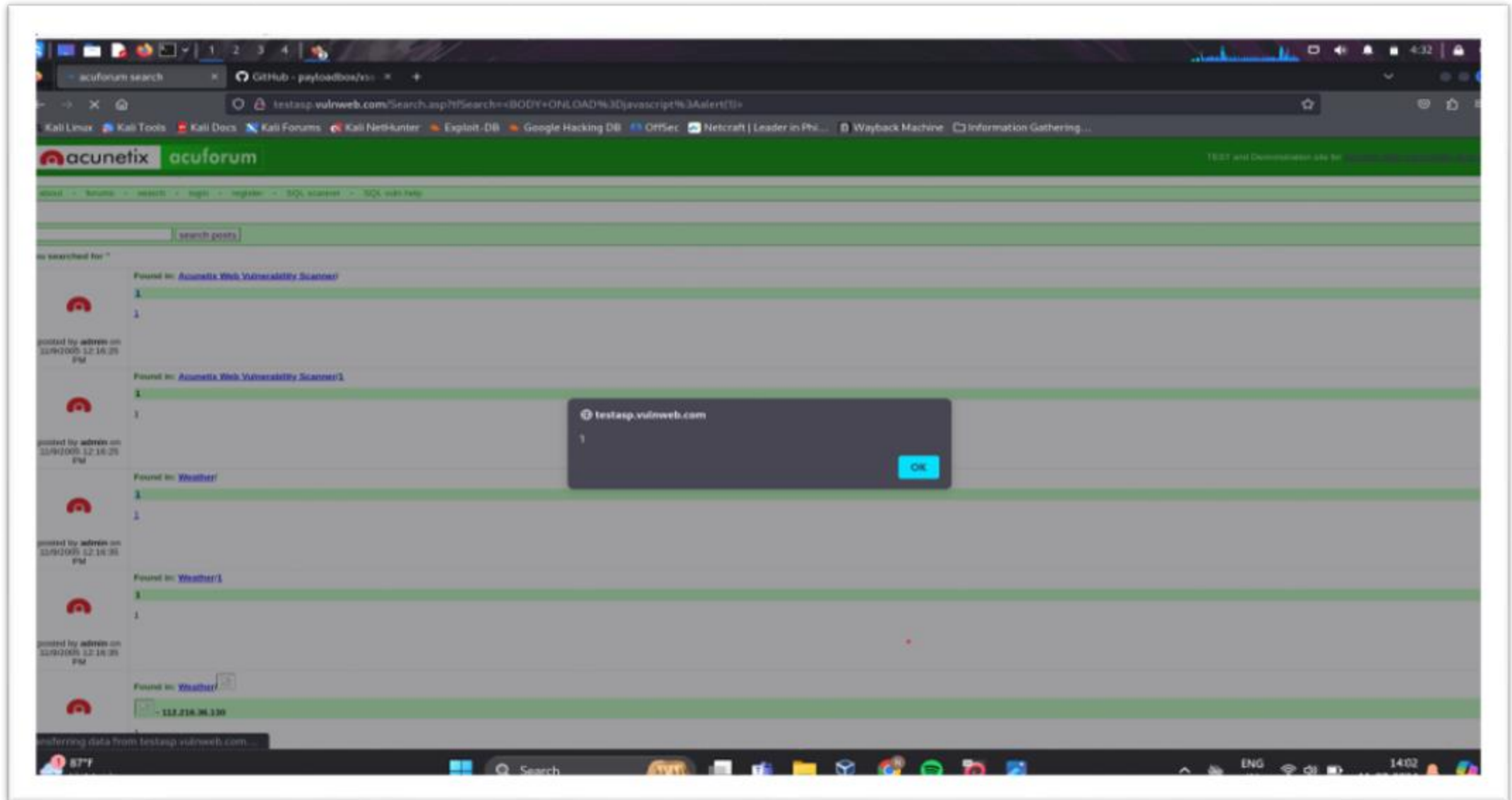
Subdomain:- <http://testasp.vulnweb.com>

**Description:** If a website directly incorporate user input into its body without proper sanitization this script wil run when the page load

**Reflected XSS** : In a reflected XSS attack the malicious script is embedded in a URL or input field.

**Impact** : Reflected XSS can lead to unauthorized access , data thef , or session hijacking. Affects users who were visit specific URL or interact with vulnerable input fields.

# 1) Proof of Concept:



## 2) Proof of concept:

kali-linux-2024.2-virtualbox-amd64 [Running] - Oracle VM VirtualBox

acuforum search GitHub - payloadbox/xss- +

testasp.vulnweb.com/Search.asp?tfSearch=<BODY+ONLOAD%3Djavascript%3Aalert(1)>






Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec Netcraft | Leader in Phi... Wayback Machine Information Gathering...

acunetix acuforum TEST and Demonstration site for [Acunetix Web Vulnerability Scanner](#)

about forums search login register SQL scanner SQL vuln help

<BODY ONLOAD=javascript: search posts

You searched for "

 posted by <b>admin</b> on 11/9/2005 12:16:25 PM	Found in: <a href="#">Acunetix Web Vulnerability Scanner/</a> 1 1
 posted by <b>admin</b> on 11/9/2005 12:16:25 PM	Found in: <a href="#">Acunetix Web Vulnerability Scanner/1</a> 1 1
 posted by <b>admin</b> on 11/9/2005 12:16:35 PM	Found in: <a href="#">Weather/</a> 1 1
 posted by <b>admin</b> on 11/9/2005 12:16:35 PM	Found in: <a href="#">Weather/1</a> 1 1
 posted by <b>aaaa</b> on	Found in: <a href="#">Weather/</a> - 112.216.36.130 1

89°F Mostly cloudy Search 14:00 11-07-2024 ENG IN

# Vulnerbilty XSS Script :

<BODY ONLOAD=javascript:alert(1)>

**Attack case and conditions:** The vulnerbilty occurs when a web application incorporates user input into its HTML content without proper sanitization or output encoding.

If the application directly reflects user input within the <body> tag the injected script runs.

# Mitigation:

- 1) Sanitize and authenticate user inputs to prevent script injection.
- 2) Always encode user-generated content before rendering it in HTML.
- 3) 3) Implement a strong Content Security Policy (CSP) to restrict Script Execution.

