Vulnerability Report :- testasp.vulnweb.com

## **Summary**

This report documents multiple security vulnerabilities found on **testasp.vulnweb.com**, a test and demonstration website. The identified vulnerabilities include **SQL Injection**, **Reflected Cross-Site Scripting (XSS)**, and **Server-Side Template Injection (SSTI)**. These security flaws could lead to **unauthorized access, sensitive data exposure, remote code execution (RCE), and potential full system compromise**. The findings are based on manual testing techniques, and each vulnerability is described in detail with steps to reproduce, impact analysis, and recommended mitigations.

## **1. SQL Injection Vulnerability**

### **Description**

The login page is vulnerable to SQL Injection. By injecting ' OR '1'='1 as the username and password, an attacker can bypass authentication.

### **Steps to Reproduce**

1. Navigate to:<http://testasp.vulnweb.com/Login.asp?RetURL=%2Fshowforum%2Easp%3Fid%3D2>
2. Enter the following credentials:
   * **Username:** ' OR '1'='1
   * **Password:** ' OR '1'='1
3. Click login.
4. The user is authenticated without valid credentials.

### **Impact**

* Unauthorized access to the system.
* Possible data exfiltration and user impersonation.

## **2. Reflected Cross-Site Scripting (XSS)**

### **Description**

The search functionality is vulnerable to Reflected XSS, allowing JavaScript execution.

### **Steps to Reproduce**

1. Visit the URL:<http://testasp.vulnweb.com/Search.asp?tfSearch=%3Cscript%3Ealert(%22HI%22)%3C/script%3E>
2. The browser executes the injected JavaScript, displaying an alert box.

### **Impact**

* Can be used for session hijacking, defacement, and phishing attacks.

## **3. Server-Side Template Injection (SSTI)**

### **Description**

Forum threads process user input without proper sanitization, leading to potential SSTI vulnerabilities.

### **Steps to Reproduce**

1. Visit the URLs:
   * <http://testasp.vulnweb.com/showthread.asp?id=3>
   * <http://testasp.vulnweb.com/showthread.asp?id=4>
   * <http://testasp.vulnweb.com/showthread.asp?id=10>
2. Inject the following payload in the thread title or message box: {{7\*7}}
3. Observe that the server interprets the input and returns 49, indicating SSTI.

### **Impact**

* Possible **Remote Code Execution (RCE)** if further exploited.
* Potential system compromise.

## **Recommendations**

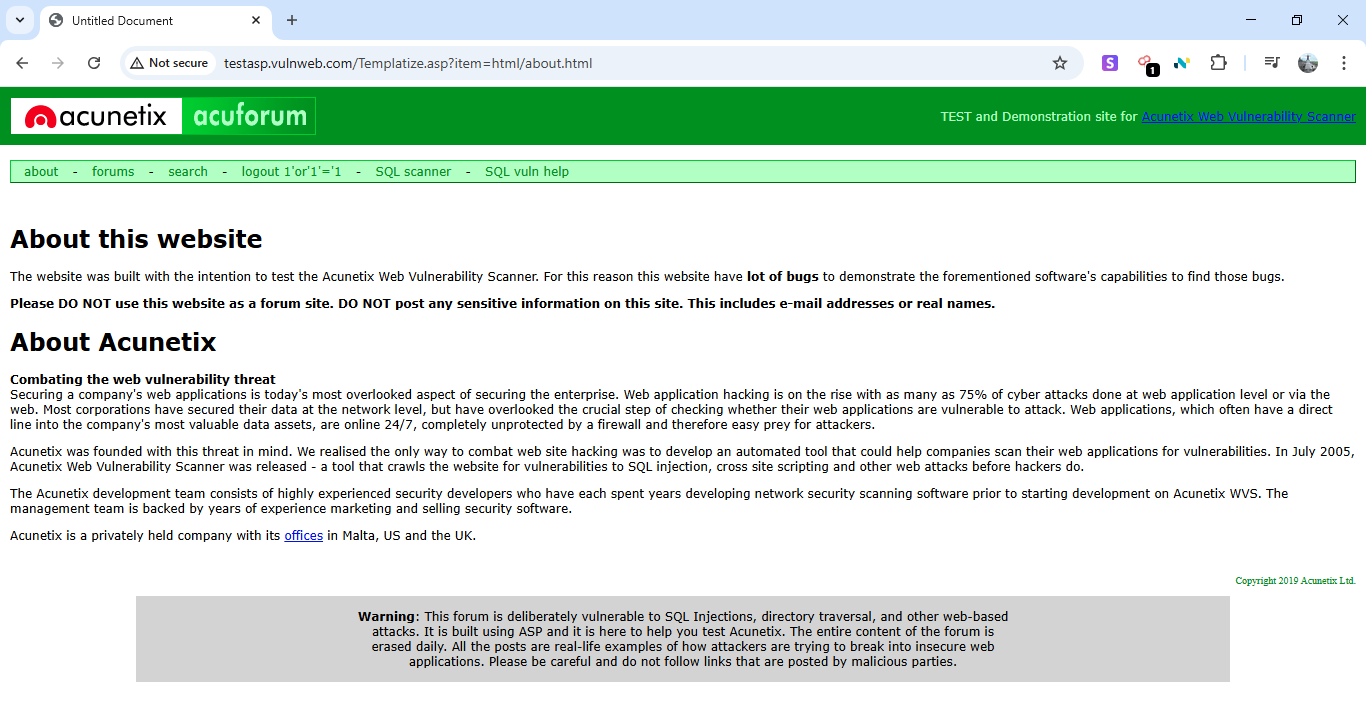
1. **SQL Injection**
   * Use **prepared statements** and **parameterized queries**.
   * Sanitize and validate user input.
2. **XSS**
   * Implement **output encoding**.
   * Use **Content Security Policy (CSP)**.
3. **SSTI**
   * Implement **strict input validation**.
   * Use a secure templating engine with sandboxing.

## **Conclusion**

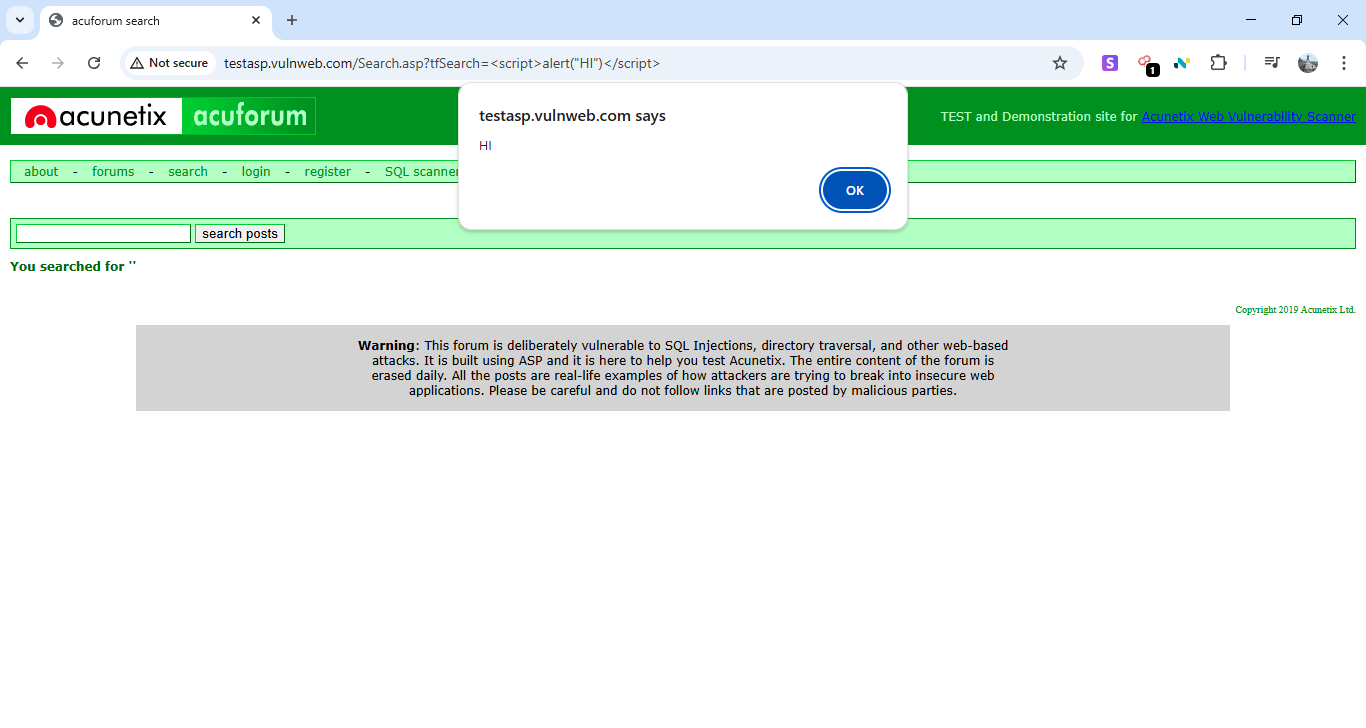
The vulnerabilities identified on **testasp.vulnweb.com** pose a significant security risk, especially in a real-world scenario. If exploited, these issues could lead to **data breaches, unauthorized access, system compromise, and reputational damage**. The recommended security measures should be implemented immediately to mitigate potential risks.

### **Evidence & Screenshots**

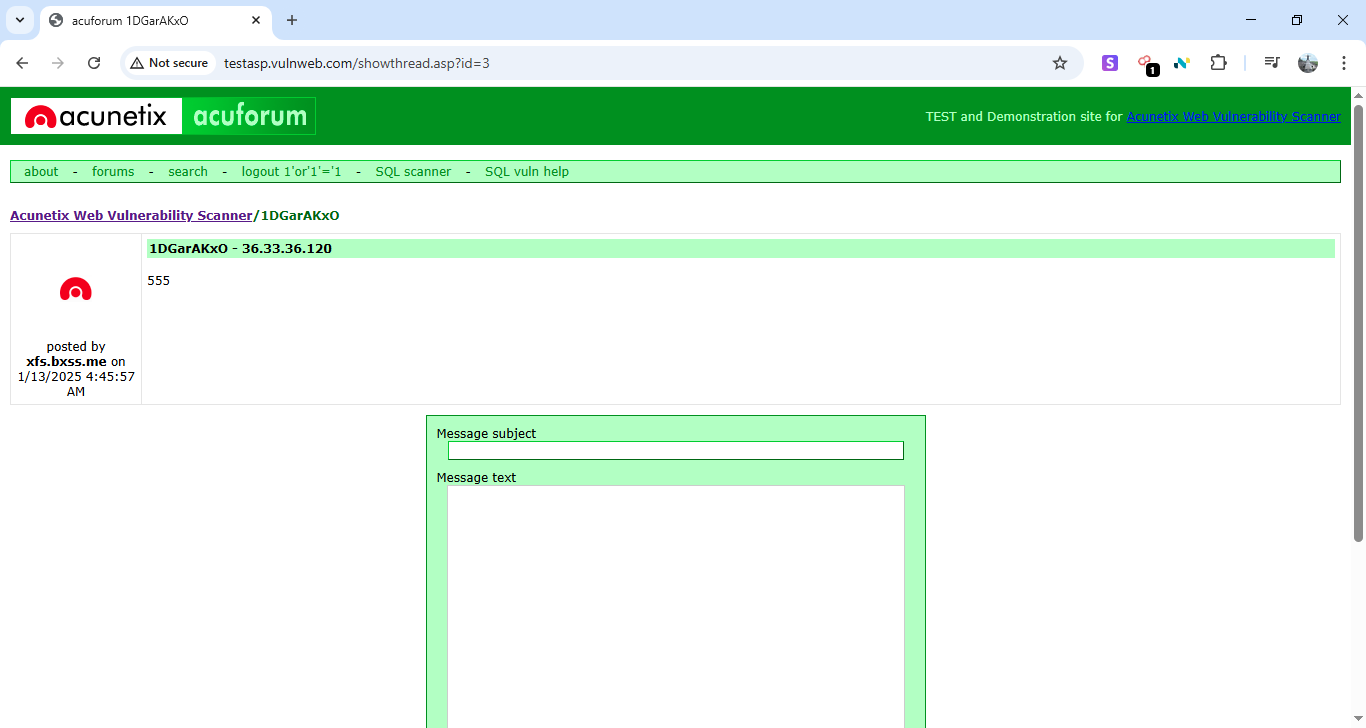
*Test 1 Screenshot :*

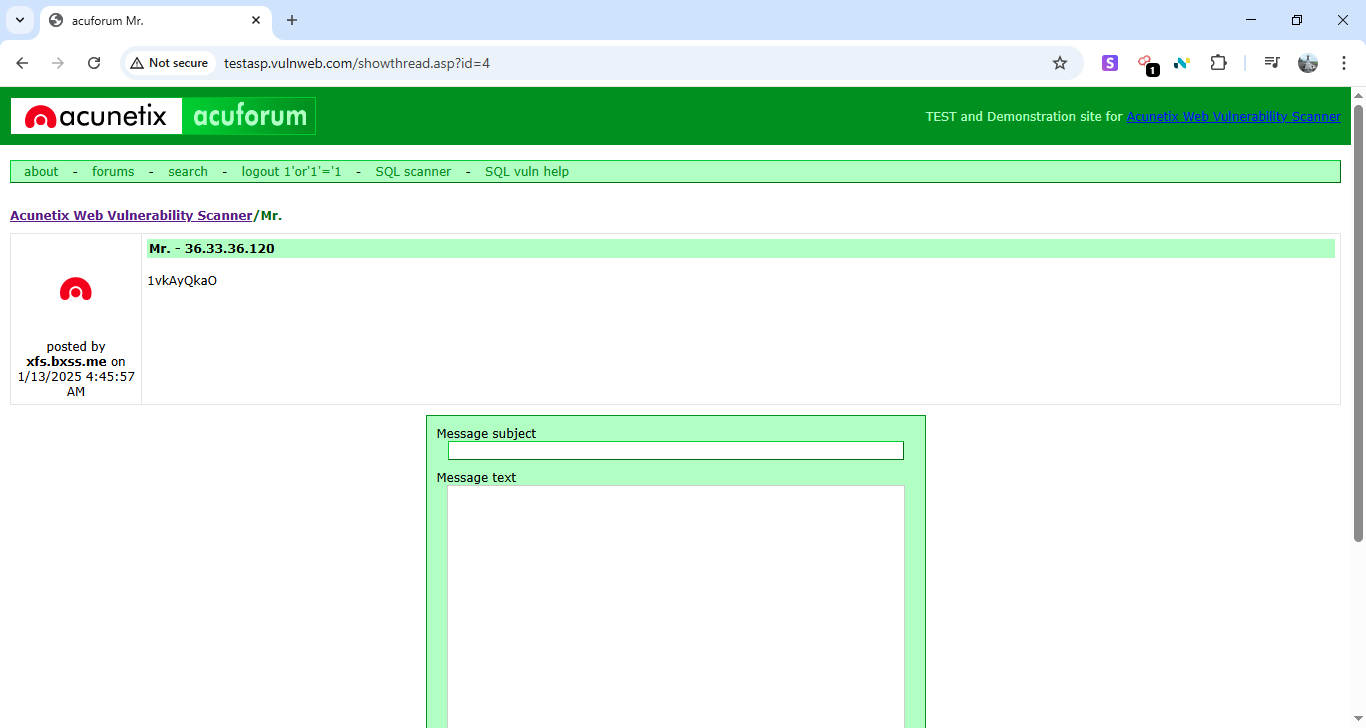
**

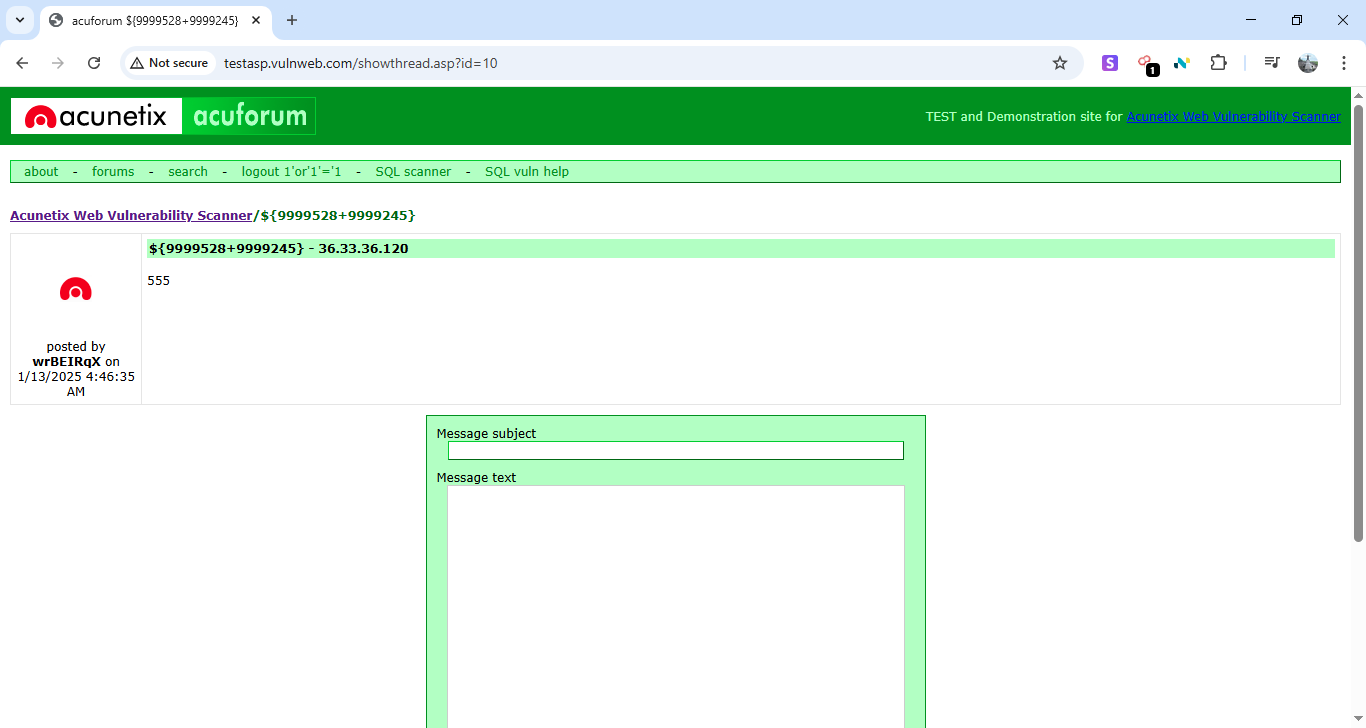
*Test 2 Screenshot :*

**

*Test 3 Screenshot :*

**

**

**