

Reporting Vulnerability

Target URL: <http://testasp.vulnweb.com/>

SQL Injection

Summary:

A boolean-based SQL injection vulnerability has been identified in the web application. This type of vulnerability allows an attacker to manipulate the application's SQL queries by injecting specially crafted input. By exploiting this vulnerability, an attacker can potentially gain unauthorized access to sensitive information or perform malicious activities.

Vulnerability Details:

Type: Boolean-based SQL Injection Application

URL: <http://testasp.vulnweb.com/>

Parameters Names: tfUName & tfUPass

Parameter Type : POST

Payload: ' or '1'='1'--

Environment:

Scope: Web Application

Product name: Acunetix(<http://testasp.vulnweb.com/>)

OS name and version (incl SP): Windows 11 22H2

Attack type: Boolean Based SQL Injection

Maximum user privileges needed to reproduce your issue: no privileges


Steps to Reproduce:

1. Navigate to the following URL: <http://testasp.vulnweb.com/Login.asp>
2. In any input field, enter the following payload: ' or '1'='1' —
3. Submit the form or proceed with the login process.

Note: After entering the provided payload (' or '1'='1'--), the affected field will always return true. This manipulation of the input causes the SQL query to evaluate the condition '1'='1' as true, bypassing any intended authentication checks and potentially granting unauthorized access to sensitive information.

Proof of Concept (PoC):

← ↻ 🏠 ⚠ Not secure | testasp.vulnweb.com/Login.asp?RetURL=%2FDefault%2Easp%3F 🔍 🔊 ☆ ☆ ⚙ InPrivate

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

[about](#) - [forums](#) - [search](#) - [login](#) - [register](#) - [SQL scanner](#) - [SQL vuln help](#)

Username:

Password:

Login

Copyright 2019 Acunetix Ltd.

Warning: This forum is deliberately vulnerable to SQL Injections, directory traversal, and other web-based attacks. It is built using ASP and it is here to help you test Acunetix. The entire content of the forum is erased daily. All the posts are real-life examples of how attackers are trying to break into insecure web applications. Please be careful and do not follow links that are posted by malicious parties.



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

[about](#) - [forums](#) - [search](#) - [logout admin](#) - [SQL scanner](#) - [SQL vuln help](#)

Forum	Threads	Posts	Last Post
Acunetix Web Vulnerability Scanner Talk about Acunetix Web Vulnerability Scanner	7	7	6/15/2023 5:31:24 PM
Weather What weather is in your town right now	1	1	11/9/2005 12:16:35 PM
Miscellaneous Anything crossing your mind can be posted here	0	0	

Copyright 2019 Acunetix Ltd.

Warning: This forum is deliberately vulnerable to SQL Injections, directory traversal, and other web-based attacks. It is built using ASP and it is here to help you test Acunetix. The entire content of the forum is erased daily. All the posts are real-life examples of how attackers are trying to break into insecure web applications. Please be careful and do not follow links that are posted by malicious parties.

Video Demonstration:

acuforum login

Not secure | testasp.vulnweb.com/Login.asp

acunetix acuforum

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

[about](#) - [forums](#) - [search](#) - [logout admin](#) - [SQL scanner](#) - [SQL vuln help](#)

Username:

Password:

Login

Copyright 2019 Acunetix Ltd.

Warning: This forum is deliberately vulnerable to SQL Injections, directory traversal, and other web-based attacks. It is built using ASP and it is here to help you test Acunetix. The entire content of the forum is erased daily. All the posts are real-life examples of how attackers are trying to break into insecure web applications. Please be careful and do not follow links that are posted by malicious parties.

Impact:

By injecting the provided payload (' or '1'='1'--), the attacker can manipulate the SQL query executed by the application. This payload leverages a boolean condition that always evaluates to true, allowing the attacker to bypass authentication mechanisms or gain unauthorized access to sensitive data stored in the application's database.

Recommendations:

- Implement proper input validation and sanitization techniques.
- Use parameterized queries or prepared statements to prevent SQL injection.
- Apply least privilege principles to database accounts used by the application.
- Secure coding practices and the risks associated with SQL injection.
- Implement a Web Application Firewall (WAF) to provide an additional layer of protection against SQL injection attacks.

Thank You

Submitted by: Vishal Balani