Home

CVE-2025-14598

CVSS: 9.8 (Critical)

SQL Injection leading to Remote Code Execution

CVE Record

Vulnerability Description:

An SQL Injection vulnerability was identified in the login functionality of BET e-Portal, a software solution widely deployed across educational institutions for student information and examination result management.

The vulnerability allowed unauthenticated attackers to interfere with backend SQL queries, potentially escalating the issue to remote code execution in certain configurations. The issue was reported and coordinated through CERT/CC, resulting in the assignment of CVE-2025-14598.

This write-up provides a technical overview of the vulnerability, its impact, how it manifested across deployments, and general remediation guidance.

Affected Products:

Product: BET e-Portal

Vendor: BeeS Software Solutions Pvt Ltd

Deployment Context:

BET e-Portal is commonly used by colleges and universities to handle student login, examination results, internal marks, and related academic workflows. It is deployed independently across 100+ institutions, each maintaining its own instance.

Vulnerability Overview:

The vulnerability is classified as SQL Injection in the login functionality. Due to insufficient input validation, user-supplied login data was incorporated into SQL queries without proper sanitization. This allowed an unauthenticated attacker to manipulate database queries and, depending on server configuration, potentially achieve remote code execution.

Impact Summary:

- Read, modify, or delete database records
- Extract sensitive student data
- Modify exam results or internal records
- Execute arbitrary SQL commands
- In certain deployments, execute OS-level commands

BET e-Portal is commonly used by colleges and universities to handle student login, examination results, internal marks, and related academic workflows. It is deployed independently across 100+ institutions, each maintaining its own instance.

Technical Details:

1. Input Vector

The vulnerability originates in the application's login form. The username and password fields were passed to the database backend in dynamically constructed SQL statements.

If user input is not strictly validated or parameterized, attackers can craft input that alters the intended SQL logic.

2. Root Cause: Unsanitized SQL Queries

The backend appears to process login attempts using SQL queries similar in structure to:

SELECT * FROM users WHERE username = '[input]' AND password = '[input]';

This pattern is vulnerable if input is not properly escaped or parameterized. Attackers can inject SQL fragments to modify query behavior.

3. Exploitation Path (Conceptual)

The general exploitation flow:

1. Attacker submits crafted input through login form
2. Application embeds this input into SQL query without sanitization
3. Database executes attacker-controlled SQL
4. Attacker gains unauthorized data access
5. Privilege escalation occurs by chaining additional SQL queries
6. In some deployments, extended procedures enabled further escalation

No special privileges are required. The attack is fully remote and unauthenticated.

4. Why Some Deployments Led to RCE

Some affected deployments had certain SQL Server features enabled, such as:

- xp_cmdshell
- Other extended stored procedures enabling system-level interaction

These features significantly amplify the impact of SQL Injection:

- SQL queries can escape the database context
- Arbitrary OS commands can be executed
- System files can be accessed or modified

Although configuration varies by institution, such misconfigurations can turn SQL injection into a remote code execution scenario.

Patch Evaluation:

Initial Remediation Attempt

During coordinated disclosure, an initial update was deployed that addressed certain input paths but did not fully resolve the vulnerability. Upon retesting, SQL injection remained possible through alternative code paths.

Final Remediation

Following further coordination through CERT/CC, additional corrective steps were implemented and verified. The final patch successfully remediated the issue across reviewed deployments

Severity & Impact

CVSS Score: 9.8 (Critical)

Vulnerability Type: SQL Injection

Potential Impact:

- **Confidentiality:** High
- **Integrity:** High
- **Availability:** High (In case of destructive queries)
- **Potential RCE:** Yes, depending on configuration

The vulnerability is considered critical in deployments where database configuration allows system-level interactions.

A CVSS vector may be added once finalized.

Mitigation Guidance:

Use Parameterized Queries / Prepared Statements

Avoid dynamically concatenating SQL strings with user input. Instead, use parameter binding mechanisms provided by frameworks or database libraries.

Enforce Strict Input Validation

Sanitize or whitelist input fields, especially those used in authentication flows.

Apply Least Privilege Principles

Ensure database accounts used by web applications have minimal permissions.

Disable High-Risk Database Features

Features such as xp_cmdshell should be disabled unless absolutely required.

Logging and Monitoring

Monitor authentication flows and SQL error logs for unusual behavior.

Regular Security Testing

Routine penetration testing and code review can identify these issues early in the development lifecycle.

Credits:

Discovered by:

Mohammed Afnaan Ahmed

www.afnaan.me

CVE ID:

CVE-2025-14598

Coordinated Disclosure:

CERT Coordination Center (CERT/CC)

References:

- CERT/CC Vulnerability Note
- CVE Record
- Researcher's Blog Post
- Researcher's Github note