

ISO 27001 Compliant Incident Management Report-SQL Injection Vulnerability

Introduction

This report details the identification and exploitation of an SQL Injection vulnerability in the Damn Vulnerable Web Application (DVWA). The test was conducted in a controlled environment to demonstrate a common vulnerability and its potential impact on application security.

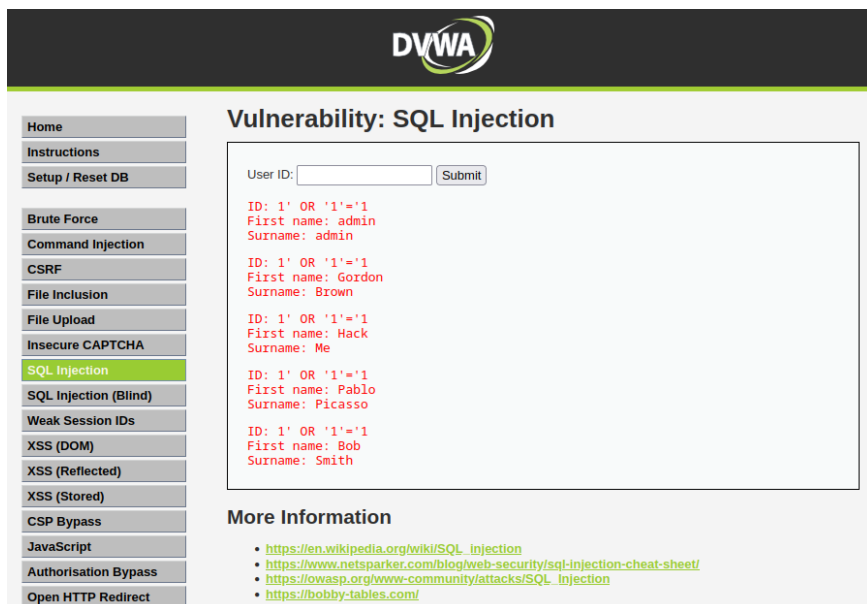
Incident description

During the security assessment of DVWA, an SQL Injection vulnerability was discovered in the “SQL Injection” module. This vulnerability allows an attacker to inject malicious SQL queries through the web application’s input fields, thereby compromising and confidentiality of the data stored in the database.

SQL Injection Method Used

To replicate and demonstrate the vulnerability, the following SQL payload was used in the “User Id” field:

1' OR '1'='1



The screenshot shows the DVWA (Damn Vulnerable Web Application) interface. The top navigation bar includes links for Home, Instructions, Setup / Reset DB, Brute Force, Command Injection, CSRF, File Inclusion, File Upload, Insecure CAPTCHA, SQL Injection (highlighted), SQL Injection (Blind), Weak Session IDs, XSS (DOM), XSS (Reflected), XSS (Stored), CSP Bypass, JavaScript, Authorisation Bypass, and Open HTTP Redirect. The main content area is titled "Vulnerability: SQL Injection" and features a "User ID:" input field with a "Submit" button. Below the input field, the results of the SQL injection are displayed in red text, showing a list of usernames and surnames: ID: 1' OR '1'='1, First name: admin, Surname: admin; ID: 1' OR '1'='1, First name: Gordon, Surname: Brown; ID: 1' OR '1'='1, First name: Hack, Surname: Me; ID: 1' OR '1'='1, First name: Pablo, Surname: Picasso; ID: 1' OR '1'='1, First name: Bob, Surname: Smith. Below the results, a "More Information" section provides links to external resources: https://en.wikipedia.org/wiki/SQL_injection, <https://www.netsparker.com/blog/web-security/sql-injection-cheat-sheet/>, https://owasp.org/www-community/attacks/SQL_injection, and <https://bobby-tables.com/>.

This payload exploits the vulnerability to modify the original SQL query in such a way that it returns all the usernames and surnames stored in the user’s table. By successfully executing this SQL Injection, the target user’s data are obtained without authorization.

Incident Impact

Exploiting this vulnerability could allow an attacker to:

- Access and extract confidential information from the database, including their name and surname.
- Modify, delete or compromise sensitive data stored in the application.

This represents a significant risk to the confidentiality, integrity and availability of the data and services provided by DVWA.

Recommendations

Based on the findings of this security assessment, the following corrective and preventive measures are recommended:

1. Input Validation: Implement strict input validations for all user-supplied data, using secure parameters in SQL queries to prevent SQL injection.
2. Penetration Testing: Conduct regular security audits, including penetration tests, to identify and mitigate security vulnerabilities before they are exploited by attackers.
3. Education and Awareness: Train technical and non-technical staff on secure application development practices and raise awareness of the risks associated with security vulnerabilities.

Conclusions

The identification and successful exploitation of the SQL injection vulnerability in DVWA underscores

the importance of proactive security in the development and maintenance of web applications.

Implementing robust security controls and following best cybersecurity practices are essential to protect critical assets and ensure business continuity.

