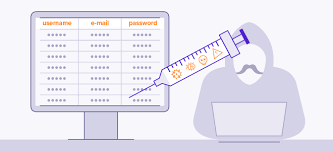
## **SQL Injection Report - OWASP BWA Mutillidae**



**By**

**RAJAN NISARGAN**

**1. Introduction**

**SQL Injection** is a type of web security vulnerability that allows attackers to interfere with the queries that an application makes to its database. It occurs when user inputs are not properly sanitized and are directly embedded into SQL queries.

Attackers can exploit SQL Injection to:

* Bypass authentication mechanisms.
* Retrieve, modify, or delete sensitive data.
* Execute administrative operations on the database.
* Potentially compromise the underlying server.

### **2. Target Application**

* **Application Name:** OWASP BWA - Mutillidae
* **Environment:** OWASP Broken Web Applications Project (OWASP BWA)

### **3. Objective**

The objective of this assessment was to detect SQL Injection vulnerabilities present within the OWASP BWA Mutillidae application and demonstrate how attackers could exploit these vulnerabilities to extract sensitive information from the database.

### **4. Attack Scenario**

* **Page Targeted:** Login Page
* **Parameter Targeted:** Username / Password Fields
* **Type of Injection:** Authentication Bypass using SQL Injection

### 

### 

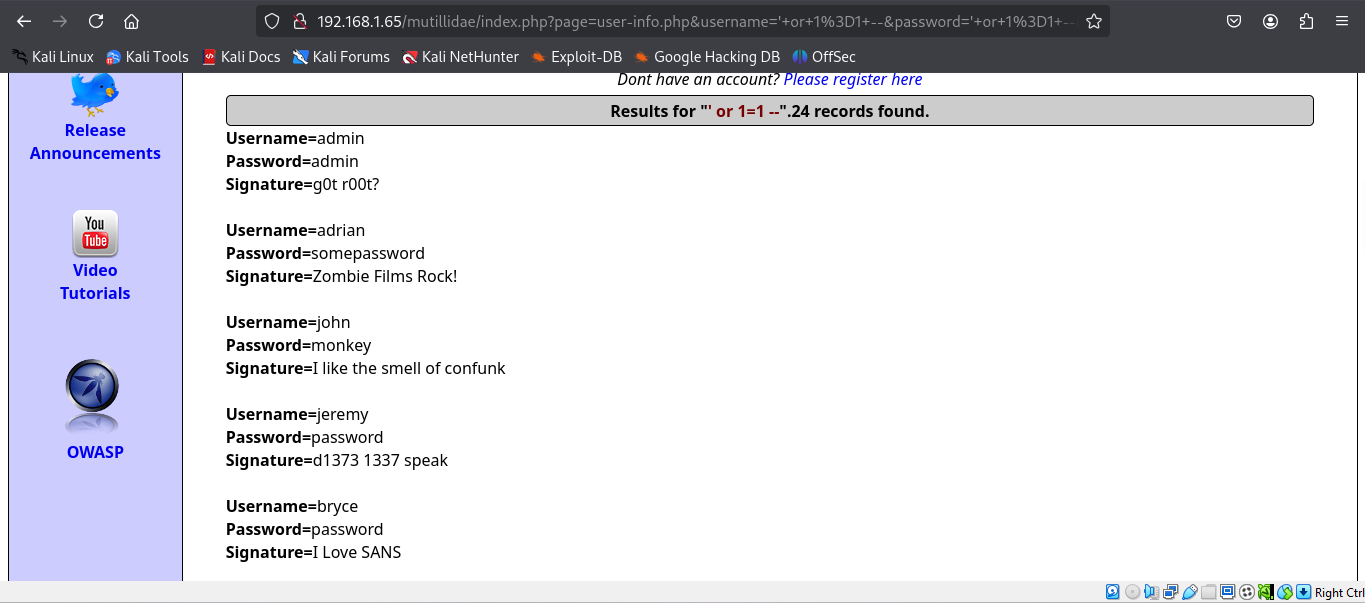
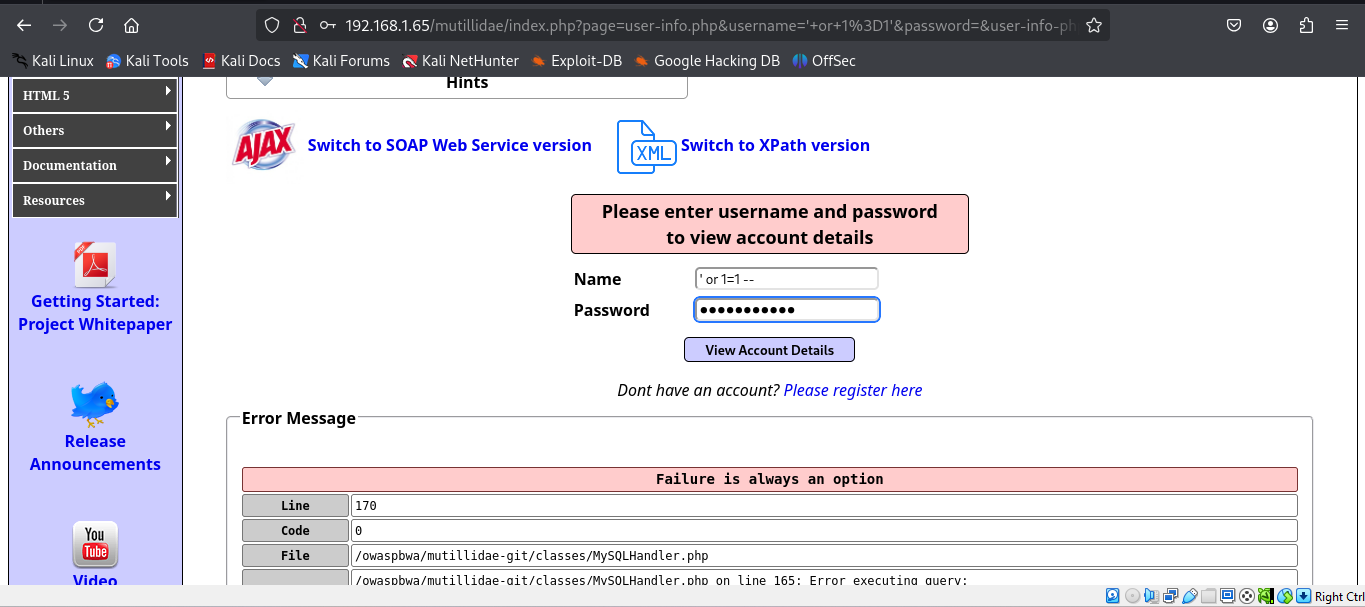
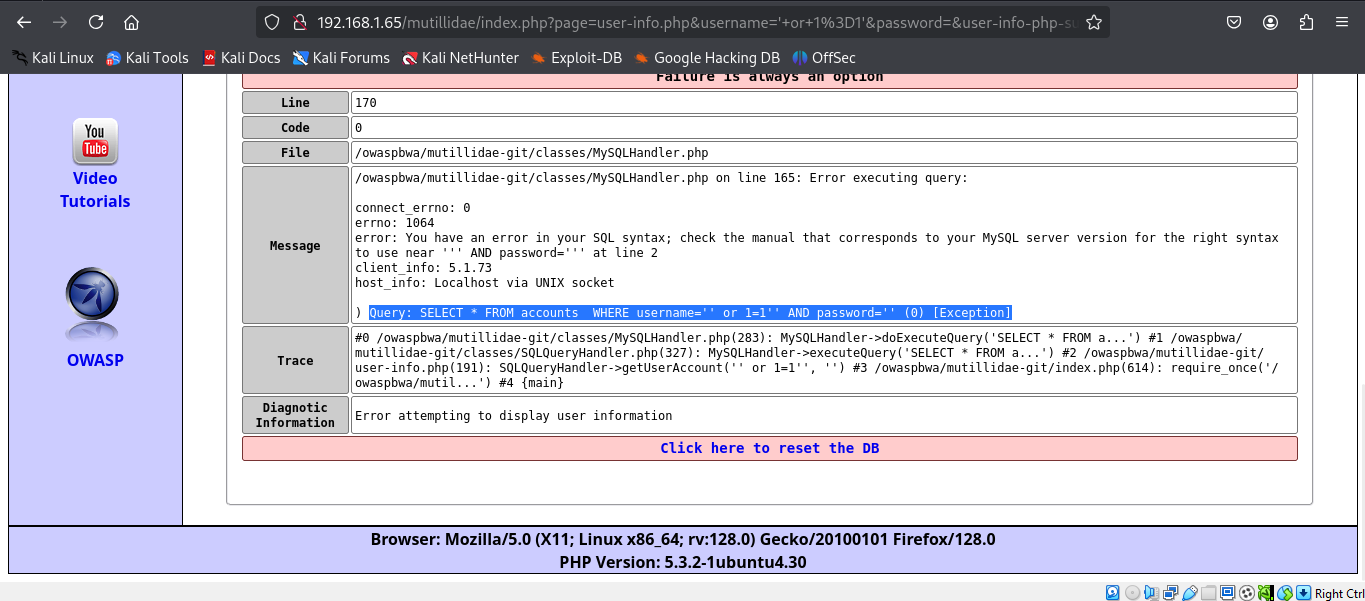
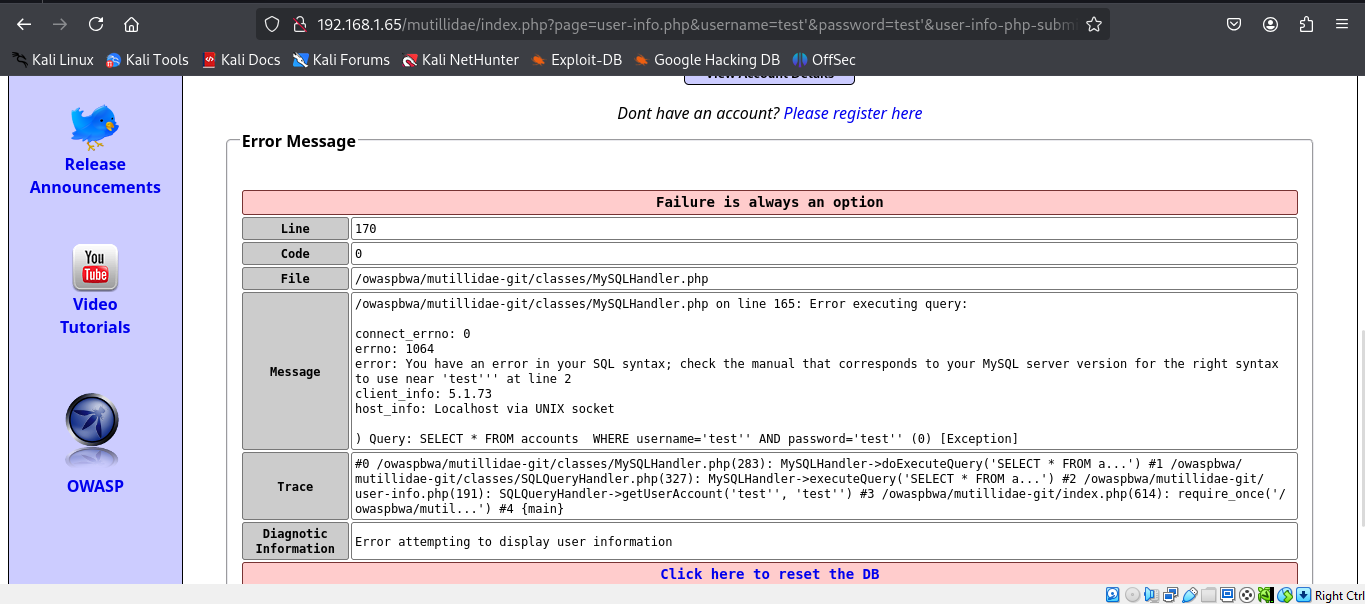
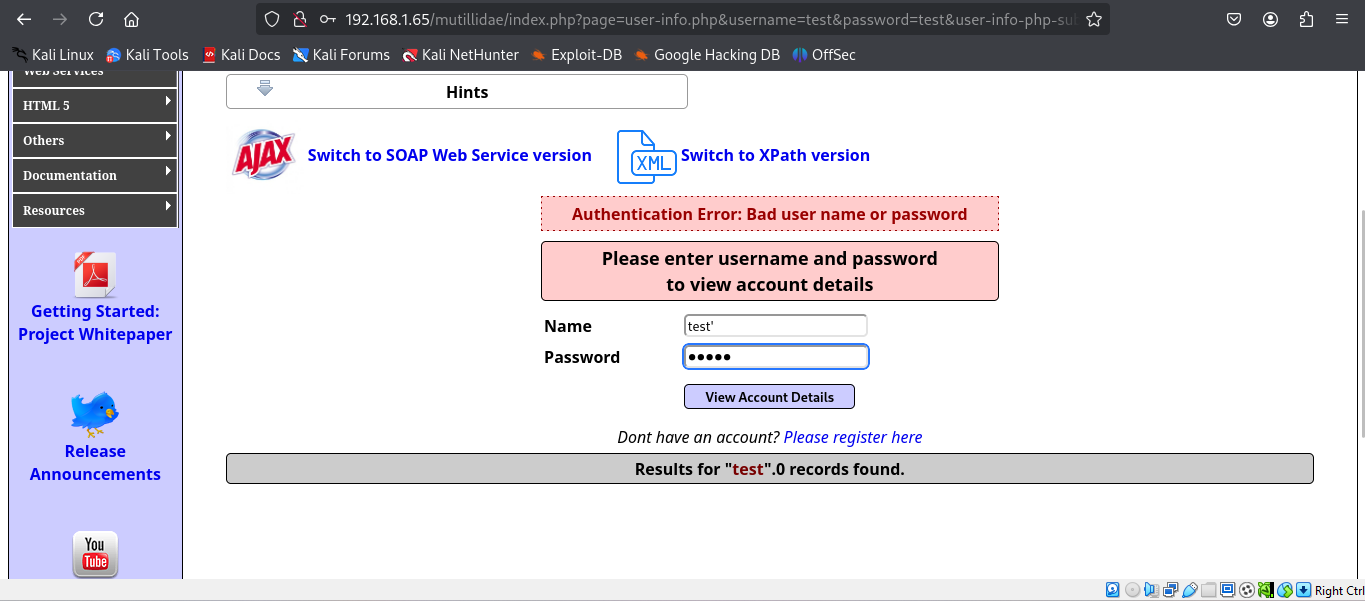
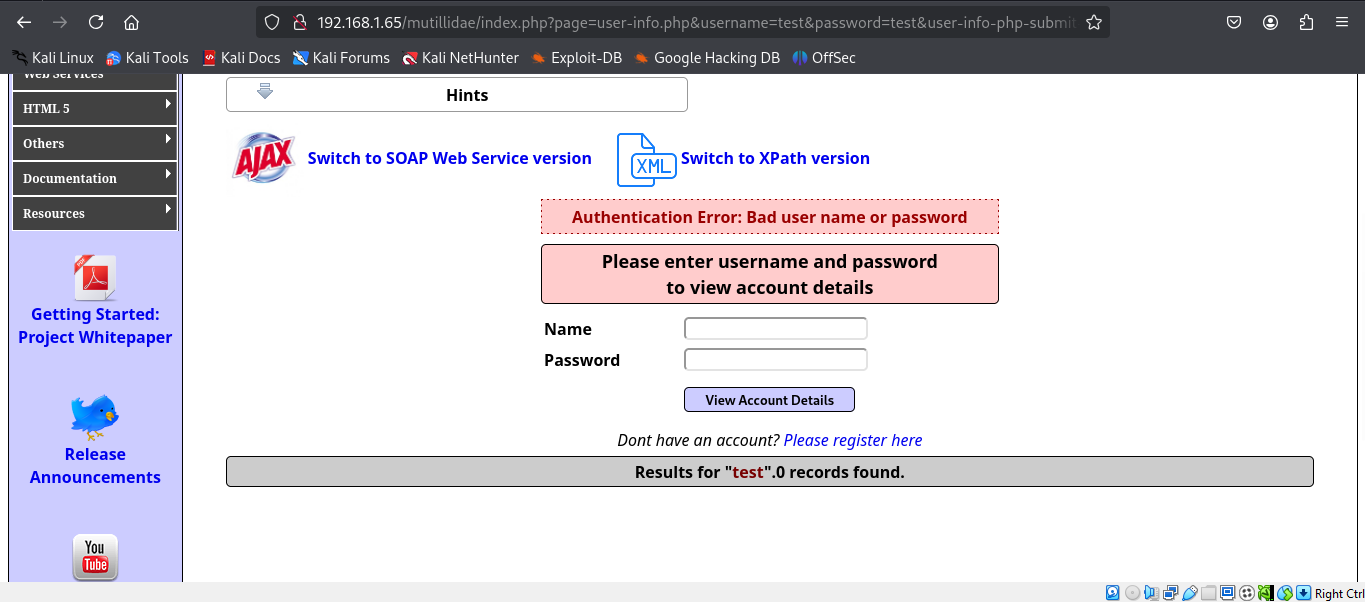
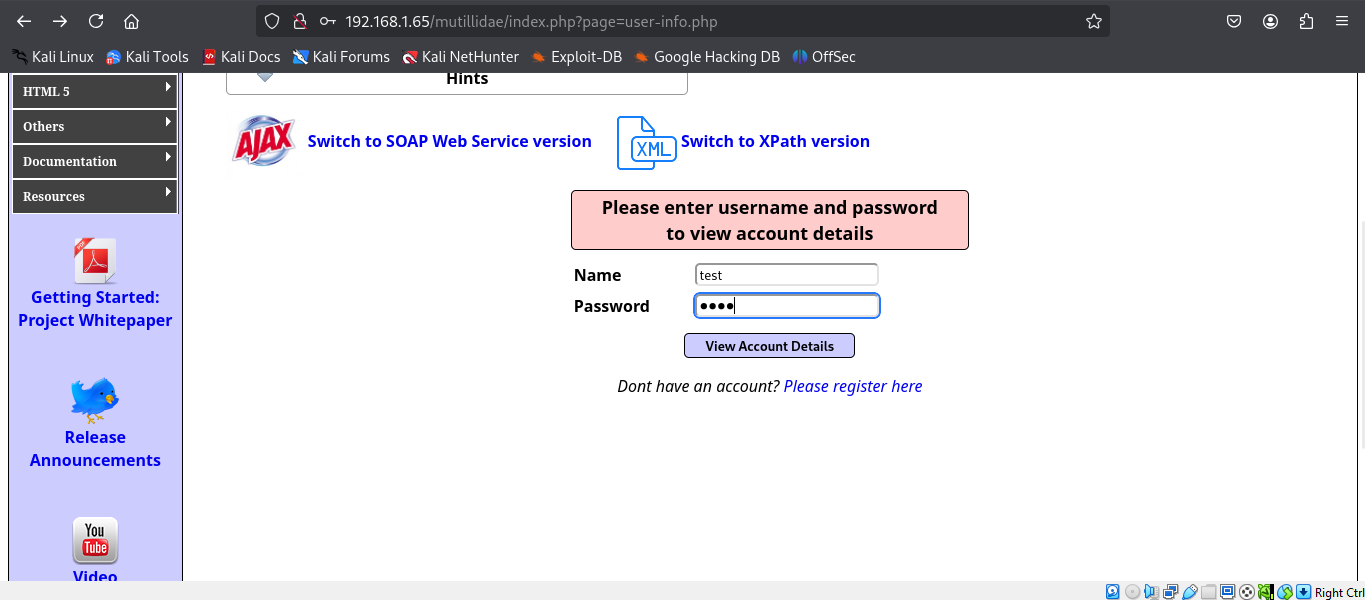
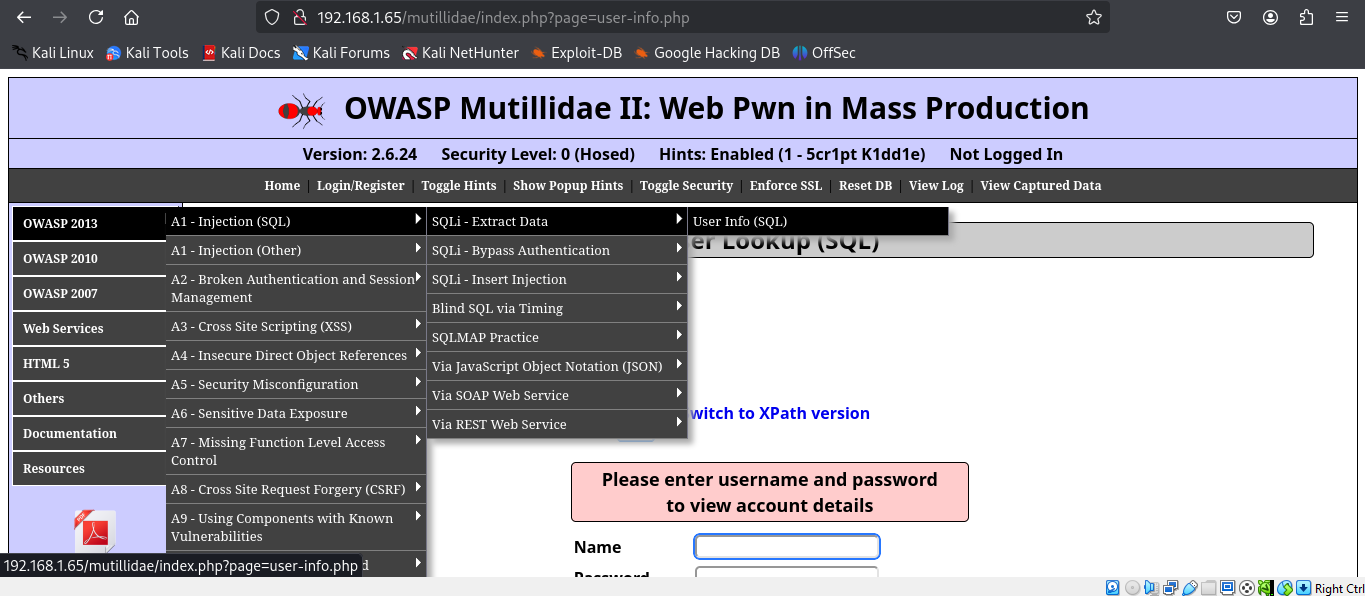
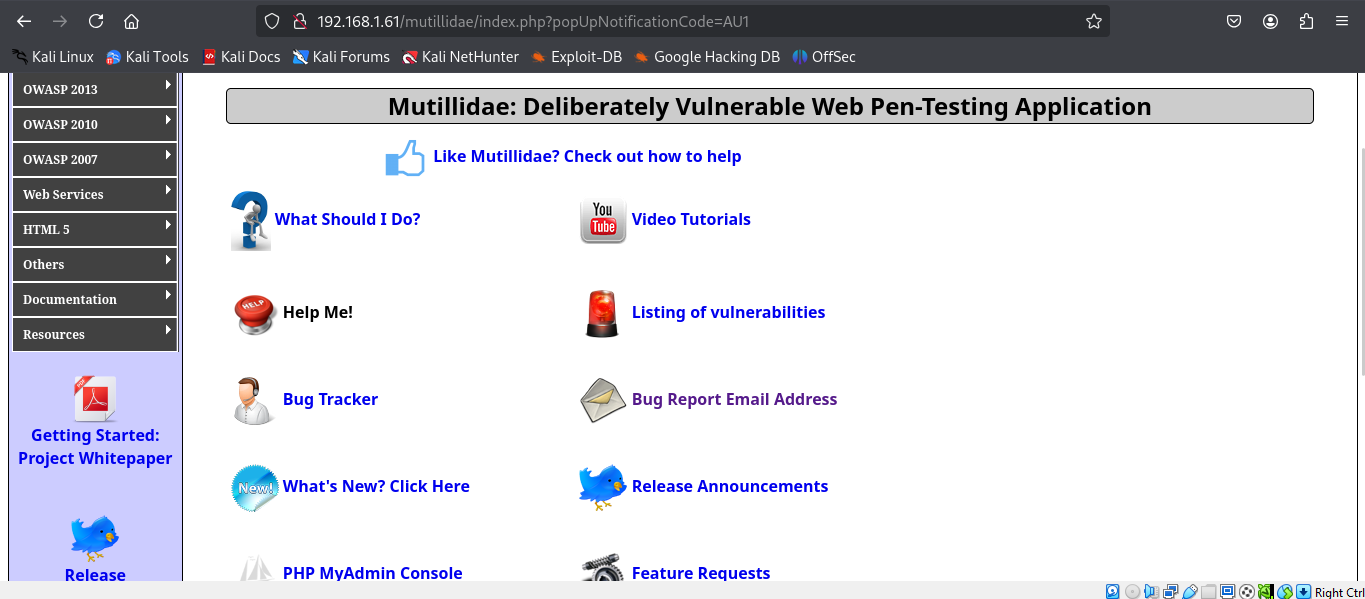
### 

### 

### 

### 

### **5. Steps Performed**



### 

### 

### **6. Exploitation**

* The payload ' OR '1'='1 modified the SQL query to always return true, thereby bypassing authentication.
* Further attempts were made to extract database information using payloads such as:
  + admin' UNION SELECT database(), version() --
  + admin' UNION SELECT table\_name, NULL FROM information\_schema.tables --

### **7. Impact**

* **Authentication Bypass:** Attackers can gain unauthorized access to sensitive areas of the application.
* **Database Information Disclosure:** Database version and schema details were extracted.
* **Data Exfiltration:** Potential risk of extracting sensitive data from various tables.

### **8. Mitigation Recommendations**

* Implement prepared statements and parameterized queries.
* Use stored procedures instead of dynamic SQL queries.
* Implement robust input validation and sanitization mechanisms.
* Apply the principle of least privilege to database accounts.
* Regularly update and patch software to fix known vulnerabilities.
* Perform regular vulnerability assessments and penetration testing.

### **9. Conclusion**

The assessment successfully demonstrated an SQL Injection vulnerability within the OWASP BWA Mutillidae application. Proper remediation techniques have been recommended to mitigate this vulnerability.

### **10. References**

* OWASP SQL Injection Prevention Cheat Sheet
* OWASP Mutillidae Documentation