

A screenshot of a web browser window. The address bar shows the URL 'webhacking.kr/challenge/web-25/'. The page title is 'SQL INJECTION'. Below the title, there is a form with two input fields: 'id : guest' and 'pw : guest'. At the bottom of the form, there are two buttons: '제출' (Submit) and '초기화' (Reset).

```
<?php
    include "../config.php";
    if($_GET['view_source']) view_source();
?><html>
<head>
<title>Challenge 50</title>
</head>
<body>
<h1>SQL INJECTION</h1>
<form method=get>
id : <input name=id value='guest'><br>
pw : <input name=pw value='guest'><br>
<input type=submit>&nbsp;&nbsp;&nbsp;&nbsp; &nbsp; &nbsp;&nbsp; &nbsp; &nbsp;&nbsp;<input type=reset>
</form>
<?php
    if($_GET['id'] && $_GET['pw']){
        $db = dbconnect();
        $_GET['id'] = addslashes($_GET['id']);
        $_GET['pw'] = addslashes($_GET['pw']);
        $_GET['id'] = mb_convert_encoding($_GET['id'],'utf-8','euc-kr');
        foreach($_GET as $ck) if(preg_match("/from|pw|#(|#)| %|=|>|<|/i",$ck)) exit();
        if(preg_match("/union/i",$_GET['id'])) exit();
        $result = mysqli_fetch_array(mysqli_query($db,"select lv from chal150 where id='{$_GET['id']}' and pw=md5('{$_GET['pw']}')"));
        if($result){
            if($result['lv']==1) echo("Level : 1<br><br>");
            if($result['lv']==2) echo("Level : 2<br><br>");
        }
        if($result['lv']=="3") solve(50);
        if(!$result) echo("Wrong");
    }
?>
<hr><a href=?view_source=1>view-source</a>
</body>
</html>
```

다른 부분들을 살펴보면 많은 필터링들이 되고 있음을 볼 수 있다. 앞선 old-27번 문제에 있었던 `preg_match` 이외에도 `if(preg_match("/union/i",$_GET['id'])) exit()`를 통해 `id`에 `union`이라는 단어가 포함되어 있으면 종료되는 등 제한이 많아 보인다.

인젝션을 해야 하는 SQL 구문을 살펴보면 다음과 같다.

```

g($_GET['id'],'utf-8','euc-kr');
tch("/from|pw|₩(|₩)| |%|=|>|<|/i",$ck)) exit();
id')) exit();
qli_query($db,"select lv from chall50 where id='{$_GET['id']}' and pw=md5('{$_GET['pw']}')");

'el : 1<br><br>");
'el : 2<br><br>");

```

일단 pw에 들어있는, md5를 이용해 암호화하는 부분을 주석처리 해야 한다. /\* \*/를 이용해서 주석처리를 하는 것이 더 수월할 것 같아 그쪽을 선택하기로 했다. 그러면 id = ' /\*~~and pw=md5~~ (\*') 이런 식으로 md5 부분이 주석처리 될 것이다.

이후 lv를 3으로 조작해줘야 하므로 뒤에 union select 3을 붙여 3을 선택해주어야 한다. 이때 id에서는 union이 필터링되므로 pw 부분에 써주어야 한다. 그렇게 하면 id = ' /\*~~and pw=md5~~ (\*' union select 3') 이런 형태가 된다. Union select 3의 뒷부분부터는 필요 없기 때문에 그 뒤에 #을 붙여 주석 처리를 해준다.

이제 남은 부분은 id의 따옴표를 닫아주는 것이다. 그런데 소스 코드에 따르면 따옴표가 필터링되므로 우회해야 한다. <https://securitynote.tistory.com/3> 를 참고해 따옴표 앞에 %aa를 붙여 필터링을 막았다. 그러면 id = ' %aa' /\*~~and pw=md5~~ (\*' union select 3 # )'와 같은 형태가 된다.

쿼리문을 url 주소창에 입력해야 하므로 #를 %23으로 써주고, 따옴표도 %27로 써주고, 공백이 필터링되므로 대신 %09를 이용한다.

그러면 최종적으로 id=%aa%27%09/\*&pw=\*/union%09select%093%23 이 된다

🔗 [https://webhacking.kr/challenge/web-25/?id=%aa%27%09/\\*&pw=\\*/union%09select%093%23](https://webhacking.kr/challenge/web-25/?id=%aa%27%09/*&pw=*/union%09select%093%23)

## SECTION

  


e/web-25/?id=%aa%27%09/\*&pw=\*/union%09select%093%23

webhacking.kr 내용:

old-50 Pwned!

확인

이렇게 문제를 해결할 수 있다.