SQL Injection Payload: ' OR '1'='1 Explained

The payload "' OR '1'='1" is a SQL Injection technique used to bypass login forms or gain unauthorized access to a system.

Here's a simple breakdown of how it works:

1. Basic SQL Query:

When a user submits a login form, the website might send a query to the database to check if the username and password are correct.

The query might look like this:

SELECT * FROM users WHERE username = '[username]' AND password = '[password]';

For example, if the user enters:

- Username: admin
- Password: password123

The query would be:

SELECT * FROM users WHERE username = 'admin' AND password = 'password123';

This query checks if the database has a user with the username 'admin' and the password 'password123'.

2. SQL Injection with "' OR '1'='1":

The malicious input "' OR '1'='1" changes the query in a harmful way.

Let's say the user enters the following in the username field:

' OR '1'='1

Now, the SQL query becomes:

SELECT * FROM users WHERE username = '' OR '1'='1' AND password = '[password]';

3. Breaking it Down:

- The username part username = '' is empty, so it doesn't match any user.

- The condition '1'='1' is always true because 1 is always equal to 1.

- Since the query checks if the username is empty or if '1 = 1', and '1 = 1' is true, the query will always return true,

bypassing the password check.

4. Result:

Since the query evaluates as true, it bypasses the password check, and the attacker is granted access, even if they don't know the correct password.

In Simple Terms:

- "' OR '1'='1" tricks the database by injecting a condition that will always be true ('1 = 1').

- This allows the attacker to bypass the password check and log in without knowing the correct credentials.

Example:

If a legitimate query looks like this:

SELECT * FROM users WHERE username = 'admin' AND password = 'password123';

An attacker might use the following input for the username field:

' OR '1'='1

This turns the query into:

SELECT * FROM users WHERE username = '' OR '1'='1' AND password = 'password123';

Since '1'='1' is always true, the query will return a valid result and allow the attacker to bypass the login.

Prevention:

To prevent such attacks, you should always use prepared statements and parameterized queries, which separate user input from SQL commands and make it impossible for attackers to inject malicious SQL.