	Hope Foundation's **Finolex Academy of Management and Technology, Ratnagiri**
	Department of Information Technology

Subject name: Advance Security Lab			Subject Code: ITL702
Class	BE	Semester –VII	Academic year: 2019-20
Name of Student			**QUIZ Score :**
Roll No		Experiment No.	02
Title: **Implement and analyze SQL Injection attack.**			

1. Lab objectives applicable: LOB1, LOB2, LOB4

2. Lab outcomes applicable: LO1, LO3

3. Learning Objectives:
1. To understand basic code injection vulnerabilities
2. To be alert about the web application attacks.

4. Practical applications of the assignment/experiment: Industries/organizations make their database dependent application SQL injection free.

5. Prerequisites:
1. SQL query formats

6. Hardware Requirements:
A Computer system with Linux/Windows OS

7. Software Requirements:
Tomcat server for JSP execution and MySql database

8. Quiz Questions (if any): (Online Exam will be taken separately batch-wise, attach the certificate/ Marks obtained)

Q1. WHAT ARE DIFFERENT SQL INJECTION STRINGS?

Q2. HOW TO PROTECT SQL INJECTION ATTACK?

9. Experiment/Assignment Evaluation:

Sr. No.	**Parameters**	**Marks obtained**	**Out of**
1	Technical Understanding (Assessment may be done based on Q & A **or** any other relevant method.) Teacher should mention the other method used -		6
2	Neatness/presentation		2
3	Punctuality		2
Date of performance (DOP)		**Total marks obtained**	**10**

Signature of the faculty

10. Theory:

What is SQL injection?

Who is vulnerable to SQL Injection attack?

List different SQL injection strings

Benefits to an attacker

Web applications that are vulnerable to SQL injection may allow an attacker to gain complete access to their underlying databases.

Demonstration of SQL Injection Attack

In demonstration we developed a website where user has to enter username and password which will be verified by server for user authentication.

Front Web page: login.html

```
<html>
<body>
<form action="http://172.16.6.144:8080/sws/login.jsp" method="post">
<h1 style="text-align:center;">Login Form</h1>
<table width="27%" align="center">
<tr><td >UserName:<input type="text" name="username"></td></tr>
<tr><td>Password:<input type="password" name="password"></td></tr>
<tr><td><input type="submit" name="submit" value="LOGIN">&nbsp;
<input type="reset" name="reset" value="RESET"/></td></tr>
</table>
</form>
</body>
</html>
////////////////////////////////////////
```

```
Jsp page for authentication:login.jsp
/////////////////////////////////////////
Int i=1;
String login, password, pin, query
username = request.getParameter("username");
password = request.getParameter("password");
Connection conn.createConnection("MyDataBase");
query = "SELECT * FROM login WHERE username='"+username+"'AND pass='"+password+"';
ResultSet result = conn.executeQuery(query);
if (result.next())
//Display User's account information
else
out.println("Login Failed");
//////////////////////////////////////////////////////
```

Now, if a user submits username and password as "famt" and "1234", the application dynamically builds and submits the query:
SELECT * FROM users WHERE username='famt' AND pass='1234'

If the username and password match the corresponding entry in the database, login successful message will be displayed to the user. If there is no match in the database,login failed message will be displayed.

Performing SQL Injection
Now, suppose an attacker submits " ' or 1=1" for the username input field (the input submitted for the password field is irrelevant). The resulting query is:
SELECT * FROM users WHERE username='' or 1=1 -- AND pass=''

The code injected in the condition (OR 1=1) transforms the entire WHERE clause into a tautology. The database uses the condition as the basis for evaluating each row. Because the conditional is a tautology,the query evaluates to true for each row in the table and returns all of them. In our example, the returned set evaluates to a non-null value, which causes the application to conclude that the user authentication was successful. Therefore, the application would show "login successful"message to an attacker.

11. Learning Outcomes Achieved

1. Understood the code injection attacks.

12. Conclusion:

1. Applications of the studied technique in industry
   a. Alert about the SQL injection vulnerability/ attack.
2. Engineering Relevance
   a. Helpful in designing a web browser addon for the SQL injection attack prevention.
3. Skills Developed
   a. Web application security.

13. References:

[1]. https://www.acunetix.com/websitesecurity/sql-injection/
[2]. https://www.guru99.com/learn-sql-injection-with-practical-example.html