Common Weakness Enumeration :
SQL INJECTION

➢ Steps for creating and running the program:

➔ Install mysql, phpmyadmin, apache2 (using refrence links present at the bottom)
➔ Create database named 'JEE_Advanced_Result_Portal' and create a results table (see results.sql in submission) in phpmyadmin.
➔ Connect the database to the program using sql query written in includes folder in connect.php
➔ Create login.php, result.php files (See comments in these files to know what each command does)
➔ login.php contains a html form which takes input from the user and redirects the page to result.php. If correct input is provided, result.php shows the fetched result from the database. Else, it redirects the page to login.php and shows "Wrong Roll Number or password" error in the header.
➔ Then, created mitigated program files (see 6th point below).
➔ Hosted this program on the local system by writing the command in terminal : php -S localhost:8000
➔ To see the webpage, go to the link localhost:8000/login.php

The above is the program which I am going to exploit by taking advantage of sql query vulnerability, that will help me to see the result of all the students without even knowing any roll no. or password. (SQL Injection).

1. Details of the weakness as YOU understand it?

I have exploited the code using SQL injection. In my php code(result.php), the following is the sql query which is vulnerable:

```
$select_query = "SELECT * FROM results WHERE roll='$roll' AND password='$password'";
```

This query fetches data from my database table (results) of the row which has roll and password as provided by the user ($roll and $password respectively) in the login page (login.php).

Now, in this query we can run another SQL command along with the given commands which can give access to the whole database without even knowing the correct roll no. and password. This can be made possible by using the fact
that the SQL query stops taking the input string at a single quote. This is WEAKNESS in the program.
And so, adding an always true condition with the query will exploit this weakness.

2. The attack model under which you have built your exploit. Is there any specific requirement for the bug to be exploited (specific operating system, hardware, libraries, etc.)?

As an attacker, I have access to login.php and result.php. The weak point of the code is the SQL query which fetches the data. It can be made to run extra command (see next point) to get access to the whole data. The only requirement for exploiting the bug is access to login.php and result.php. The exploiting will be done by providing an intelligent Input.

3. How to run your vulnerable system to expose the exploit.

- ➔ Firstly, write the command : php -S localhost:8000 in the termianal to connect the login.php and result.php pages to the database and start the session.
- ➔ Go to the webpage : localhost:8000/login.php

- ➔ Provide the input:

Roll no. : 123' or '1'='1' or '1'='1

- ➔ Here, 123 can be anything and of any length
- ➔ **Password : abc**
- ➔ Here, abc can be anything and of any length

THIS WILL FLASH THE COMPLETE RESULT OF EVERY STUDENT

OR

- ➔ Provide the input:

Roll no. : 123

- ➔ Here, 123 can be anything and of any length

- ➔ **Password : abc' or '1'='1**
- ➔ Here, abc can be anything and of any length

THIS WILL FLASH THE COMPLETE RESULT OF EVERY STUDENT

4. Explain how your exploit works (in the accompanying program/system).

My exploit takes advantage of SQL query vulnerability in the following way-

A SQL query takes input which is enclosed in two single quotes.
So, in the query
$select_query = "SELECT * FROM results WHERE roll='$roll' AND password='$password'";
In roll='$roll', the first single quote starts to take the user input roll no. Now, when provide an input:

123' or '1'='1' or '1'='1

the sql query takes the roll only till the single quote after 3. Now, 'or' starts new command in the same query and checks if the condition '1'='1' is true or not. As this condition is always true, the SQL query gets green signal for fetching all the data even if roll no and password are wrong.

The query becomes:

A or B or (C and D)

where A is roll, B is '1'='1' and C is '1'='1

If any one of these becomes true then the query fetches the whole data.

Similarly for password,

the query becomes:

A and B or C

where A=roll, B=password and C='1'='1 which is always true and hence fetches all the data.

5. Explain how to mitigate this weakness, i.e. exactly pin-point what was wrong in the program/system and what is the way the program should have been written.

To mitigate this weakness, I have use the command:

mysqli_real_escape_string()

This command escapes special characters in a string for use in an SQL query, taking into account the current character set of the connection. So, even if there is a single quote in the user input, the sql query does not stop taking the input at the single quote since it is escaped. Hence the whole input is taken as a roll no. or password.
So in the mitigated login.php and result.php, I have just made this slight change in the sql query which stops giving the access to the result.

References:

https://www.liquidweb.com/kb/install-apache-2-ubuntu-18-04/ ----> link for installing apache2 and enabling php
https://www.liquidweb.com/kb/install-phpmyadmin-ubuntu-18-04/ -----> link to install phpmyadmin
https://www.digitalocean.com/community/tutorials/how-to-install-mysql-on-ubuntu-18-04 -----> installing mysql
https://cwe.mitre.org/top25/archive/2020/2020_cwe_top25.html
https://samate.nist.gov/SRD/index.php
https://www.w3schools.com/sql/sql_injection.asp
https://www.w3schools.com/php/