**A PROJECT REPORT ON**

**DETECTION AND PREVENTION**

**OF SQL INJECTION**

SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE

IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE AWARD OF THE DEGREE

OF

**BACHELOR OF ENGINEERING (COMPUTER ENGINEERING)**

##### SUBMITTED BY

**VISHAL TANAWADE 18CS055**

**SARTHAK THORAT 18CS056**

**HITESH SASWADKAR 18CS048**

**SURAJ SHIRUDE 18CS051**

**PRATIK KARHEKAR 18CS027**



## DEPARTMENT OF COMPUTER ENGINEERING

## AISSMS COLLEGE OF ENGINEERING

PUNE 411001

## SAVITRIBAI PHULE PUNE UNIVERSITY

## 2021 -2022



**CERTIFICATE**

This is to certify that the project report entitles

“**SECURED LOGIN USING SQL INJECTION”**

Submitted by

**VISHAL TANAWADE 18CS055**

**SARTHAK THORAT 18CS056**

**HITESH SASWADKAR 18CS048**

**SURAJ SHIRUDE 18CS051**

**PRATIK KARHEKAR 18CS027**

is a bonafide student of this institute and the work has been carried out by him/her under the supervision of **Prof. S.S.Jadhav** and it is approved for the partial fulfillment of the requirement of Savitribai Phule Pune University, for the award of the degree of **Bachelor of Engineering** (Computer Engineering).

**Prof. S.S. Jadhav** **Dr. D. P. Gaikwad**

Guide Head,

Department of Computer Engineering Department of Computer Engineering

Place : Pune

Date :

**ACKNOWLEDGEMENT**

We take this opportunity to thank Head Of The Department for giving us an opportunity to work on a mini-project that helped hone our technical skills to a great extent. We are immensely grateful to **Prof. S. S. Jadhav** for their guidance, support and suggestions that helped us in making this project a success.

It has been an honor to be guided by **Prof. S. S. Jadhav** for her terrific mentorship, and we are very thankful for all of her constructive criticism that greatly helped us in our journey towards the successful completion of the project.

We would also like to acknowledge the efforts and appreciate the departmental faculty for their efforts.

**INDEX**

Contents:

1 Introduction

1.1 Problem Statement

1.2 Objectives

2 Methodology

2.1Modeling

2.2 Implementation and Evaluation

3 Result

4 Conclusion

1. **INTRODUCTION**

The Internet and web applications are the modern day workplace and business ground contributing to driving the world economy. At the same time, hackers and attackers have developed a parallel underground economy of hacking into web applications and stealing a large amount of sensitive business-critical information with malicious intent. Among the various security threats a web application is exposed to, SQL Injection Attack (SQLIA) and cross-site scripting (XSS) has taken the forefront. It has prevailed as a popular attack method. Using a SQL injection attack, an attacker can extract, modify or destroy the back end database of web applications. The simplicity of attacking a web application using SQL injection and abundance of vulnerable applications on the Internet has largely contributed to widespread data breach incidents. Every web app developer should consider web vulnerabilities as much as possible with full effort in order to avoid SQLI and XSS vulnerabilities. Our systems detects SQL injection attacks on web servers. For detection of these attacks we use policies. Policies are basically regular expressions. This regular expressions detects request with exploits. SQL injection attacks are done on databases

of web server.

* 1. **PROBLEMSTATEMENT**

To create a secured login via PHP and SQL and prevents SQL Injection of any level.

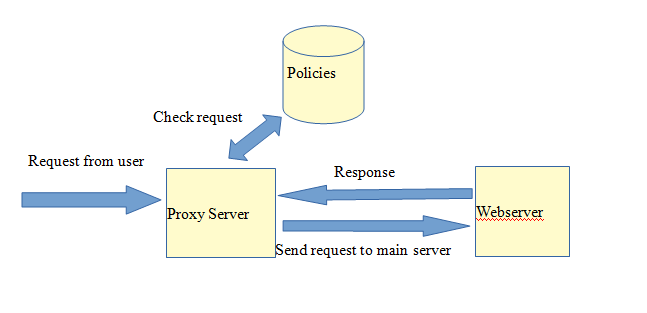
## 1.2OBJECTIVE

With the world switching to remote work on a scale never seen previously, cybercriminals have become more active than ever. Security at many organizations has suffered since workers have started working from insecure home networks and using their own (possibly infected) personal computers. As a result, the potential danger from the most frequent attack vectors can hardly be over estimate.

**2.METHODOLOGY**

* Primary Defenses:
* Option 1: Use of Prepared Statements (with Parameterized Queries)
* Option 2: Use of Properly Constructed Stored Procedures
* Option 3: Allow-list Input Validation
* Option 4: Escaping All User Supplied Input
* Additional Defenses:
* Also: Enforcing Least Privilege
* Also: Performing Allow-list Input Validation as a Secondary Defense

**2.1 Modeling:**



**2.2 Implementation and Evaluation:**

We have developed a proxy which helps in detection and avoiding SQL injection and cross site scripting attack. A website that is purposefully vulnerable to SQL injection attacks is developed. The login page is vulnerable to SQL injection making it easy to bypass login. The system then detects the SQL Injection and we can prevent that login using below function:

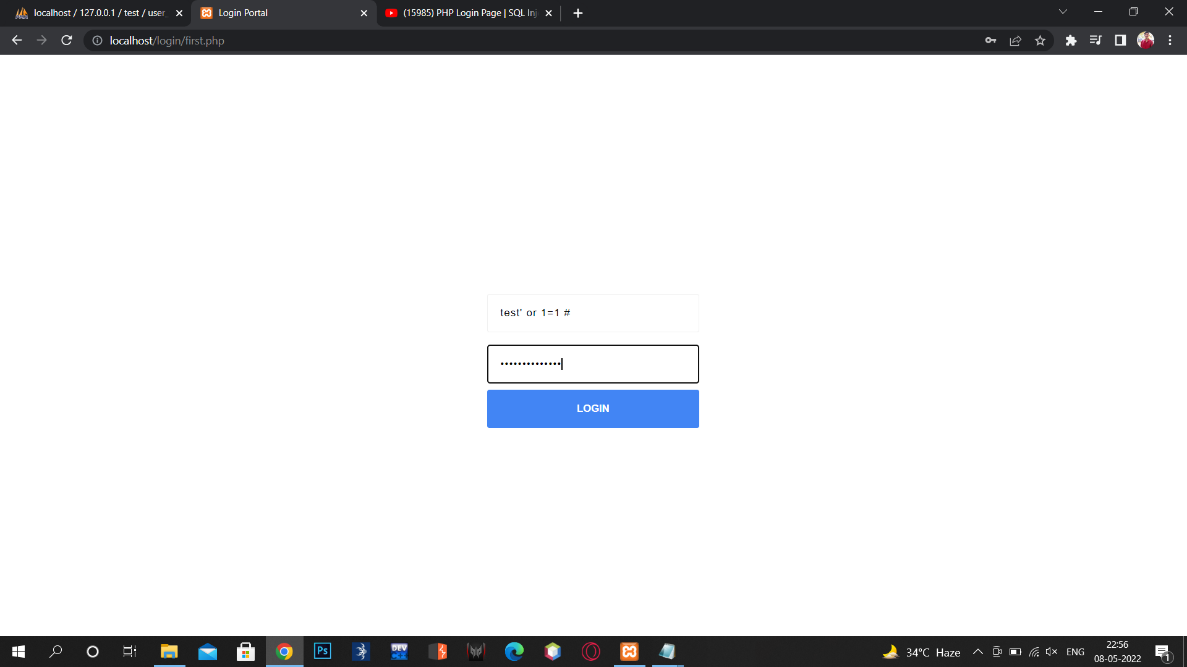
**mysqli\_real\_escape\_string()**

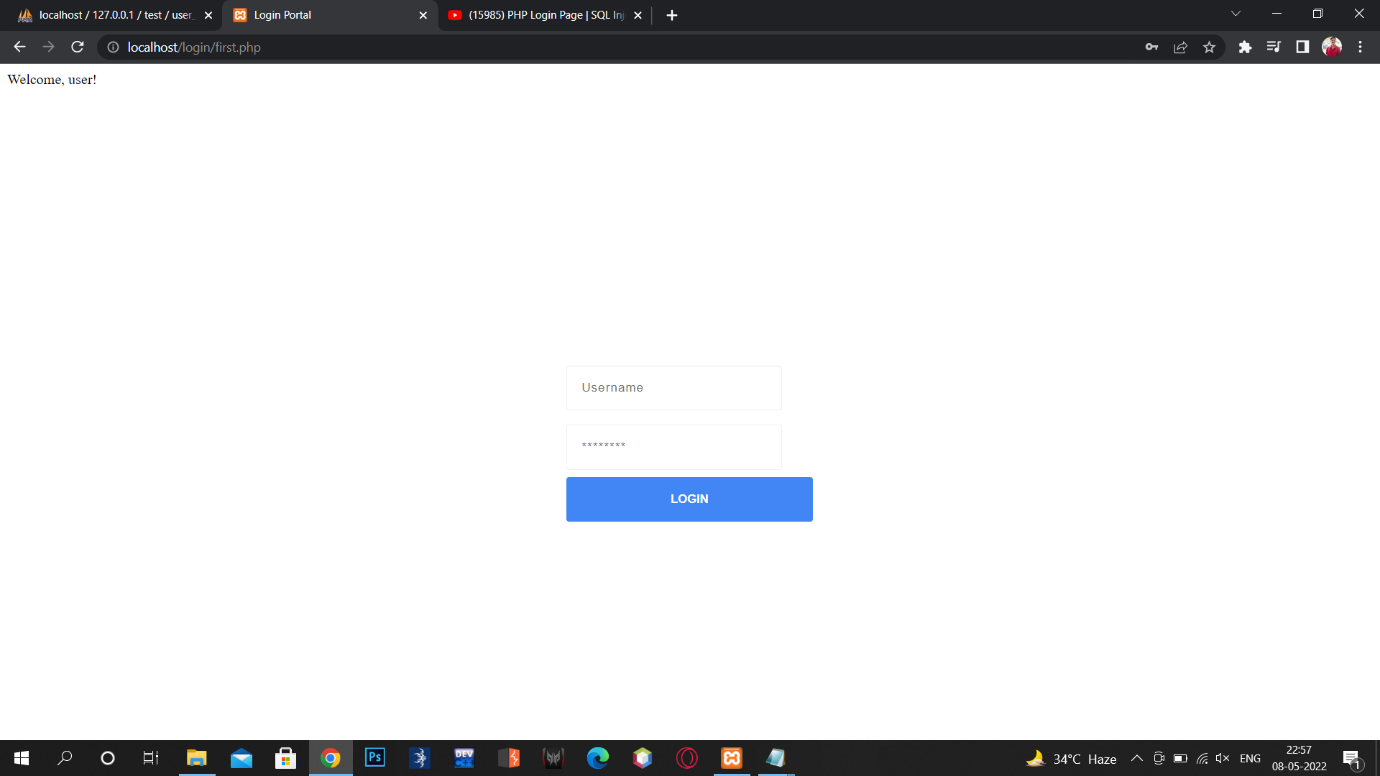
It special characters in a string for use in an SQL statement, taking into account the current charset of the connection

## Parameter Values

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| connection | Required. Specifies the MySQL connection to use |
| escapestring | Required. The string to be escaped. Characters encoded are NUL (ASCII 0), \n, \r, \, ', ", and Control-Z. |

**3.RESULT**

****

****

**4.CONCLUSION**

Thus, our project can help in detecting and preventing SQL injection attacks. Also it can be extended to detect and prevent other attacks on web server.