A Mini Project Report

On

"Secured and Unsecured Site Demonstration Using SQL Injection And Cross Scripting"

Submitted to the

Savitribai Phule Pune University

In partial fulfillment of Laboratory Practice-03

By

Shubham Kokane(BECOMPA-35)

Shreyas Soni(BECOMPA-68)

Piyush Wadi(BECOMPA-72)

Under the guidance of

Prof. N. B. Korade

Department Of Computer Engineering

Pimpri Chinchwad Education Trust's

Pimpri Chinchwad College of Engineering & Research

Ravet, Pune-412101

(2020-2021)

PIMPRI CHINCHWAD EDUCATION TRUST'S

Pimpri Chinchwad College of Engineering and Research, Pune 412101

CERTIFICATE

This is to certify that *(Shubham Kokane, Shreyas Soni, Piyush Wadi)* has successfully completed the project entitled "**Secured and Unsecured site demonstration using SQL Injection and Cross Scripting**" in the fulfillment of B.E. (Computer Engineering), and this work has been carried out in my presence.

Place:

Date:

Prof.N. B. Korade
Guide,
Department of Computer Engineering,
Pimpri Chinchwad College of Engineering and Research,
Pune-412101

Prof. Dr. Archana Chaugule
(HOD), Department of Computer Engineering,
Pimpri Chinchwad College of Engineering and Research,
Pune-412101

Prof. Dr. H.U. Tiwari
Principal,
Pimpri Chinchwad College of Engineering and Research,
Pune-412101

ACKNOWLEDGEMENT

This is a great pleasure & immense satisfaction to express my deepest sense of gratitude & thanks to everyone who has directly or indirectly helped me in completing my seminar work successfully.

I express my gratitude towards Project guide Prof. N. B. Korade and Prof. Dr.Archana Chaugule, Head of Department of Computer Engineering, Pimpri Chinchwad college of Engineering and Research, Pune 412101 who guided & encouraged me in completing the seminar work in scheduled time. I would like to thank our Principal (Prof. Dr. Tiwari H.U.), for allowing us to pursue my seminar in this institute.

No words are sufficient to express my gratitude to our parents for their unwavering encouragement. We also thank all my friends for being a constant source of my support.

Shubham Kokane
Shreyas Soni
Piyush Wadi

INDEX

LIST OF FIGURES

1.INTRODUCTION

1.1 Network Security

Network security consists of the policies and practices adopted to prevent and monitor unauthorized access, misuse, modification, or denial of a computer network and network-accessible resources. Network security involves the authorization of access to data in a network, which is controlled by the network administrator. Users choose or are assigned an ID and password or other authenticating information that allows them access to information and programs within their authority. Network security covers a variety of computer networks, both public and private, that are used in everyday jobs; conducting transactions and communications among businesses, government agencies and individuals. Networks can be private, such as within a company, and others which might be open to public access. Network security is involved in organizations, enterprises, and other types of institutions. It does as its title explains: It secures the network, as well as protecting and overseeing operations being done. The most common and simple way of protecting a network resource is by assigning it a unique name and a corresponding password.

1.2 Types of attacks

Networks are subject to attacks from malicious sources. Attacks can be from two categories: "Passive" when a network intruder intercepts data traveling through the network, and "Active" in which an intruder initiates commands to disrupt the network's normal operation or to conduct reconnaissance and lateral movement to find and gain access to assets available via the network

Types of attacks include :

1.2.1 Active attacks:

An Active attack attempts to alter system resources or affect their operations. Active attacks involve some modification of the data stream or creation of false statements. Types of active attacks are as following:

- **Masquerade –**

  Masquerade attacks take place when one entity pretends to be a different entity. A Masquerade attack involves one of the other forms of active attacks.

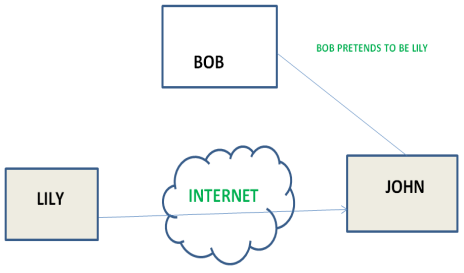

Fig 1:.Masquerade Attack Example

- **Modification of messages –**

  It means that some portion of a message is altered or that message is delayed or reordered to produce an unauthorized effect. For example, a message meaning “Allow JOHN to read confidential file X” is modified as “Allow Smith to read confidential file X”.

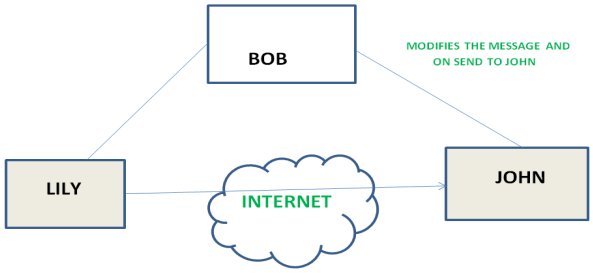

Fig 2: Modification Of Messages

- **Repudiation –**

  This attack is done by either sender or receiver. The sender or receiver can deny later that he/she has sent or received a message. For example, a customer asks his Bank “To transfer an amount to someone” and later on the sender(customer) denies that he had made such a request. This is repudiation.

- **Replay –**

  It involves the passive capture of a message and its subsequent transmission to produce an authorized effect.

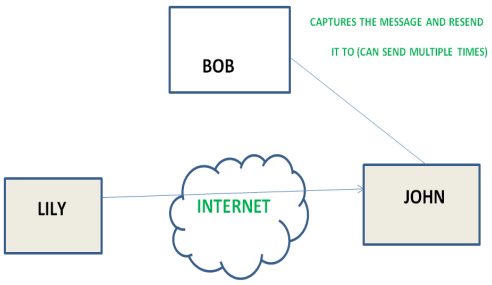

Fig 3: Replay Attack

- **Denial of Service –**

  It prevents normal use of communication facilities. This attack may have a specific target. For example, an entity may suppress all messages directed to a particular destination.

Another form of service denial is the disruption of an entire network either by disabling the network or by overloading it by messages so as to degrade performance.

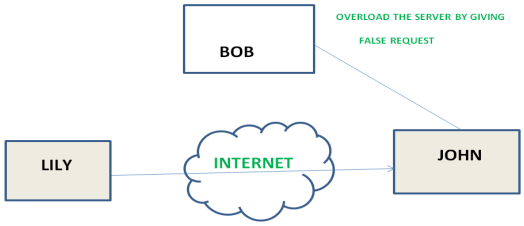

Fig 4: Denial of Service Attack

1.2.2 Passive attacks:

A Passive attack attempts to learn or make use of information from the system but does not affect system resources. Passive Attacks are in the nature of eavesdropping on or monitoring of transmission. The goal of the opponent is to obtain information that is being transmitted. Types of Passive attacks are as following:

- **The release of message content –**

  Telephonic conversation, an electronic mail message or a transferred file may contain sensitive or confidential information. We would like to prevent an opponent from learning the contents of these transmissions.

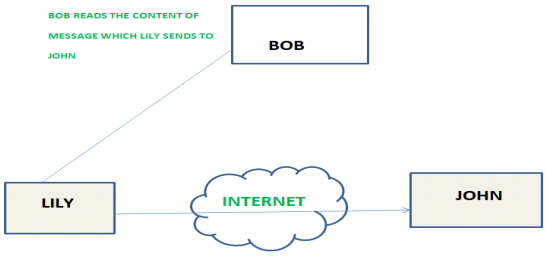

Fig 5: The release of message contents

- **Traffic analysis –**

  Suppose that we had a way of masking (encryption) of information, so that the attacker, even if captured the message, could not extract any information from the message.
  The opponent could determine the location and identity of the communicating host and could observe the frequency and length of messages being exchanged. This information might be useful in guessing the nature of the communication that was taking place

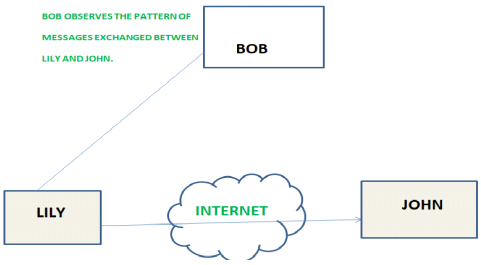

Fig 6: Traffic Analysis

2. SQL INJECTION ATTACK

SQL injection is a code injection technique, used to attack data-driven applications, in which diabolical SQL statements are inserted into an entry field for execution (e.g. to dump the database contents to the attacker). SQL injection must exploit a security vulnerability in an application's software, for example, when user input is either incorrectly filtered for string literal escape characters embedded in SQL statements or user input is not strongly typed and unexpectedly executed. SQL injection is mostly known as an attack vector for websites but can be used to attack any type of SQL database.

SQL injection attacks allow attackers to spoof identity, tamper with existing data, cause repudiation issues such as voiding transactions or changing balances, allow the complete disclosure of all data on the system, destroy the data or make it otherwise unavailable, and become administrators of the database server.

Form

SQL injection (SQLI) was considered one of the top 10 web application vulnerabilities of 2007 and 2010 by the Open Web Application Security Project. In 2013, SQLI was rated the number one attack on the OWASP top ten. There are four main sub-classes of SQL injection:

- Classic SQLI
- Blind or Inference SQL injection
- Database management system-specific SQLI
- Compounded SQLI
    - SQL injection + insufficient authentication
    - SQL injection + DDoS attacks
    - SQL injection + DNS hijacking
    - SQL injection + XSS

This SQL code is designed to pull up the records of the specified username from its table of users. However, if the "userName" variable is crafted in a specific way by a malicious user, the SQL statement may do more than the code author intended. For example, setting the "userName" variable as:

```
' OR '1'='1
```

or using comments to even block the rest of the query (there are three types of SQL comments). All three lines have a space at the end:

```
' OR '1'='1' --
' OR '1'='1' {
' OR '1'='1' /*
```

SQLIA is a hacking technique in which the attacker adds SQL statements through a web application's input fields or hidden parameters to access resources. Lack of input validation in web applications causes hackers to be successful. For the following examples we will assume that a web application receives a HTTP request from a client as input and generates a SQL statement as output for the back end database server.

For example an administrator will be authenticated after typing: employee id=112 and password=admin. Figure 1 describes a login by a malicious user exploiting SQL Injection vulnerability. Basically it is structured in three phases:

1.an attacker sends the malicious HTTP request to the web application
2.creates the SQL statement
3.submits the SQL statement to the back end database

Figure 1: Example of a SQL Injection Attack

The above SQL statement is always true because of the Boolean tautology we appended (OR 1=1) so, we will access the web application as an administrator without knowing the right password.

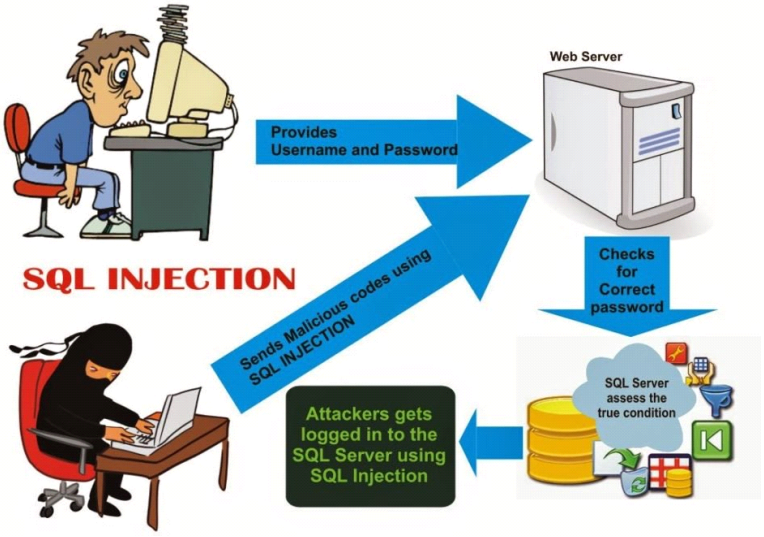

Fig 7 : SQL Injection Attack Example

3.Cross Scripting

Cross-site scripting (XSS) is a type of security vulnerability typically found in web applications. XSS attacks enable attackers to inject client-side scripts into web pages viewed by other users. A cross-site scripting vulnerability may be used by attackers to bypass access controls such as the same-origin policy. Cross-site scripting carried out on websites accounted for roughly 84% of all security vulnerabilities documented by Symantec up until 2007.[1] XSS effects vary in range from a petty nuisance to significant security risk, depending on the sensitivity of the data handled by the vulnerable site and the nature of any security mitigation implemented by the site's owner network.

4.PROBLEM STATEMENT

SQL Injection attacks and Cross-Site Scripting attacks are the two most common attacks on web applications. Develop a new policy-based Proxy Agent, which classifies the request as a scripted request or query-based request, and then, detects the respective type of attack, if any in the request. It should detect both SQL injection attacks as well as Cross-Site Scripting attacks.

5.REQUIREMENTS

Hardware Requirements:

- 4GB RAM
- 500GB HardDisk
- Windows Machine

Software Requirements:

- PHP
- XAMPP
- MySQL

6. SCREENSHOTS:

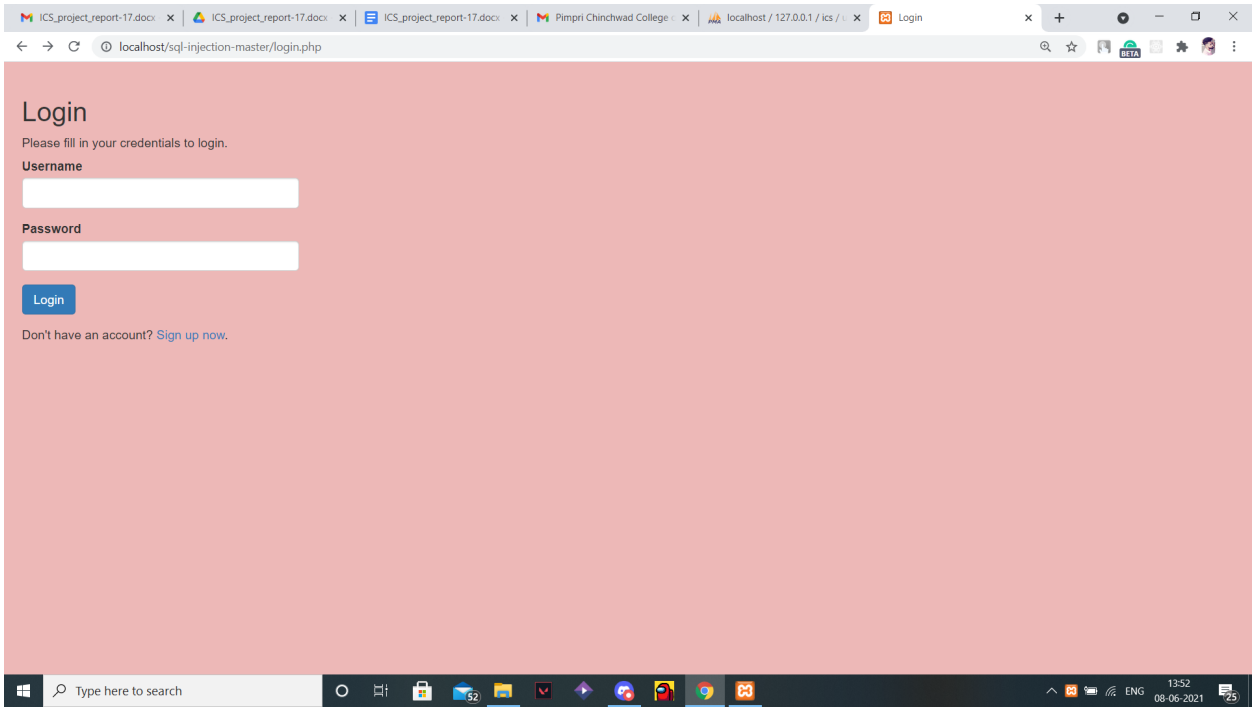

Fig8: Admin Login page

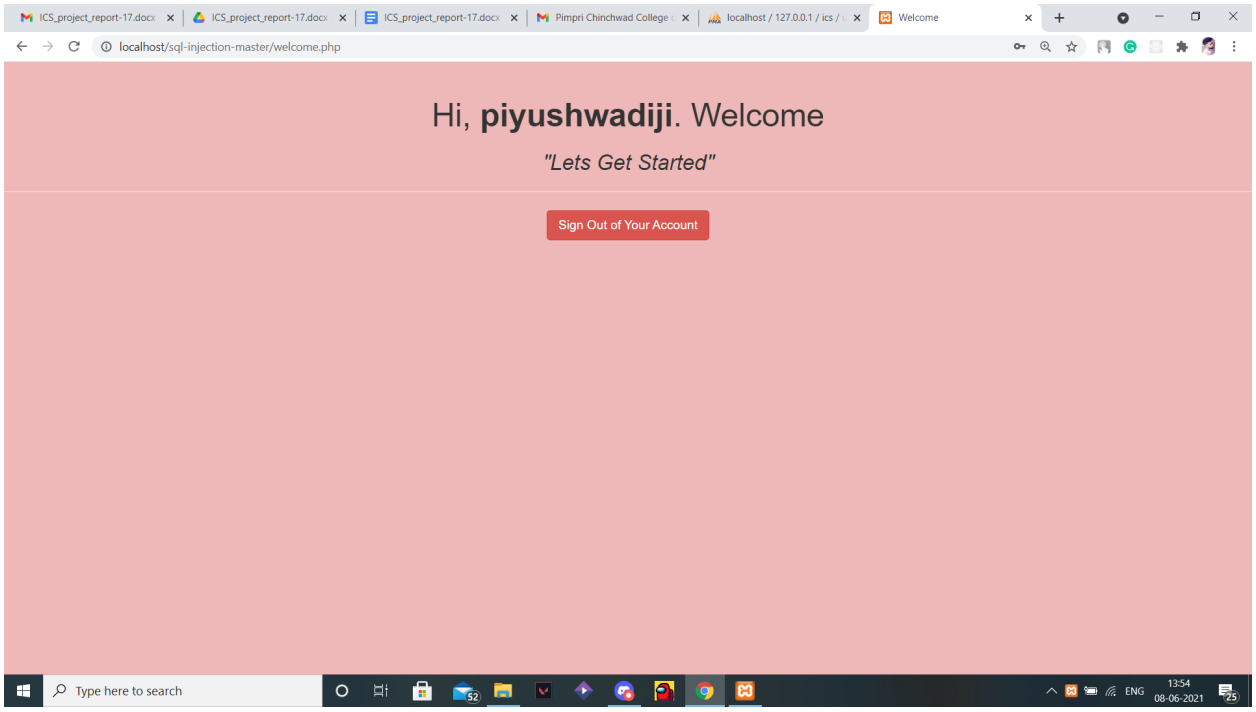

Fig 9: Admin Login success page

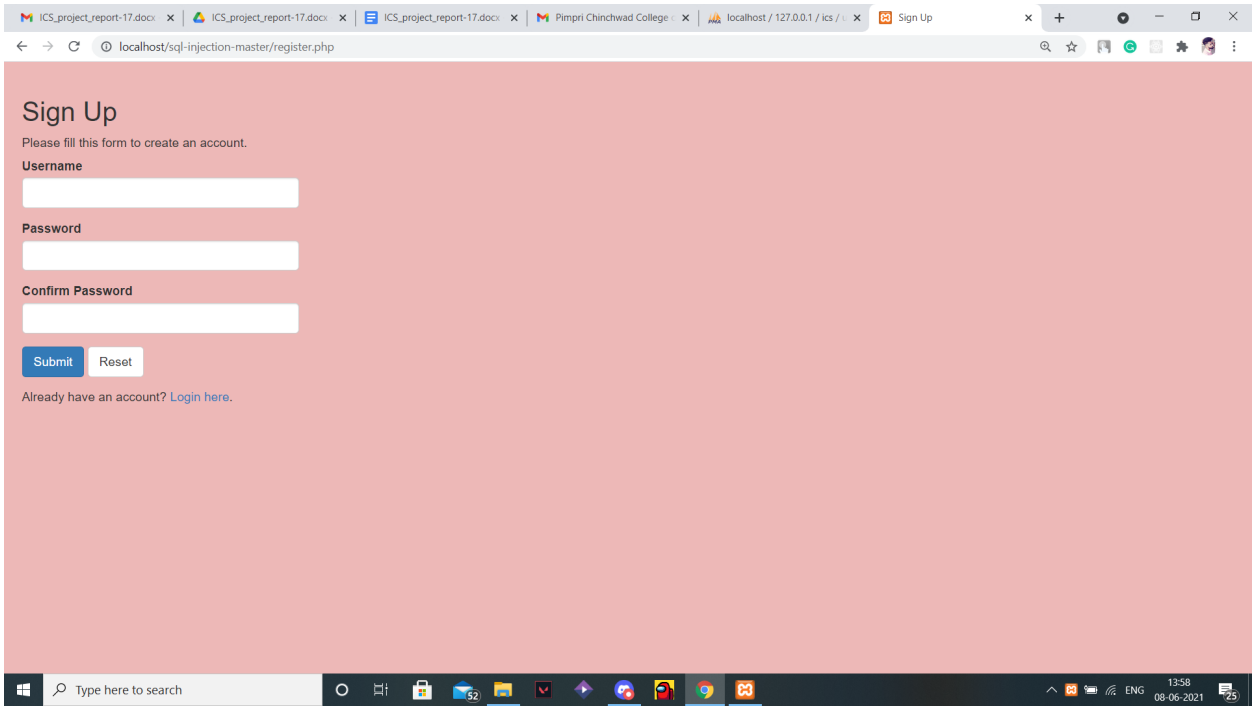

Fig10: Signup Page

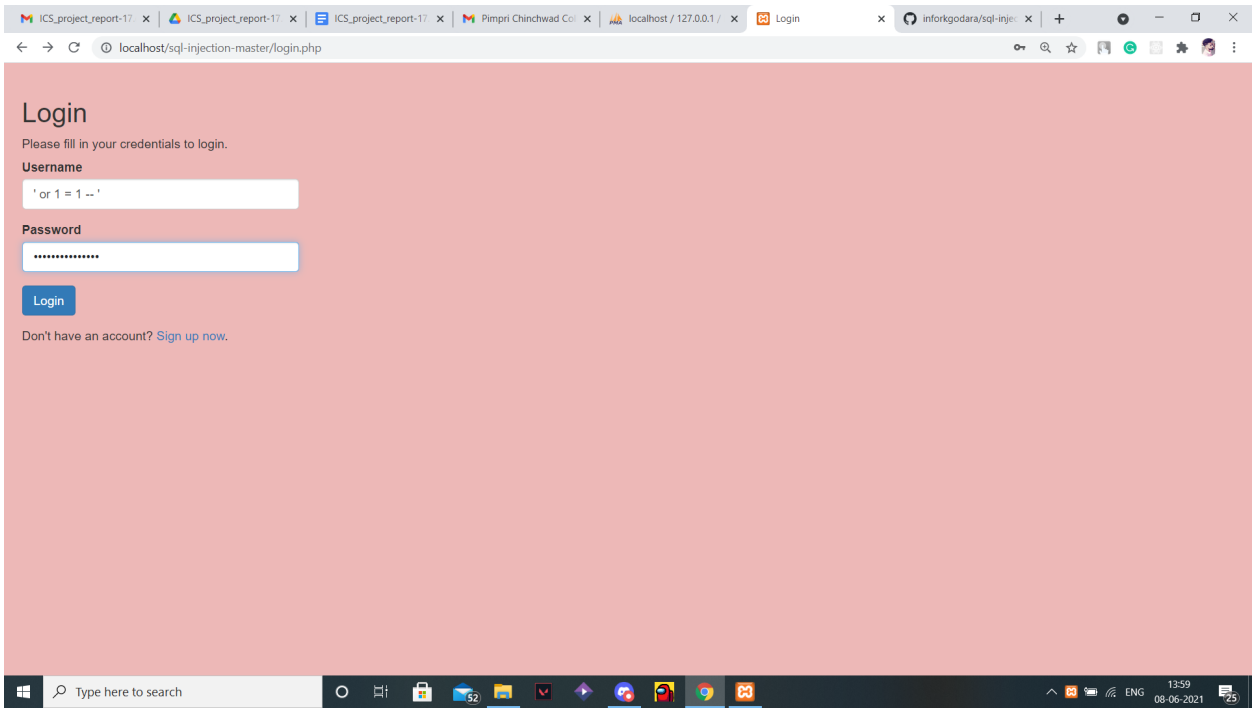

Fig12: SQL Injection attempt on secure login

Fig13: Error notification after SQL Injection attempt on secure login

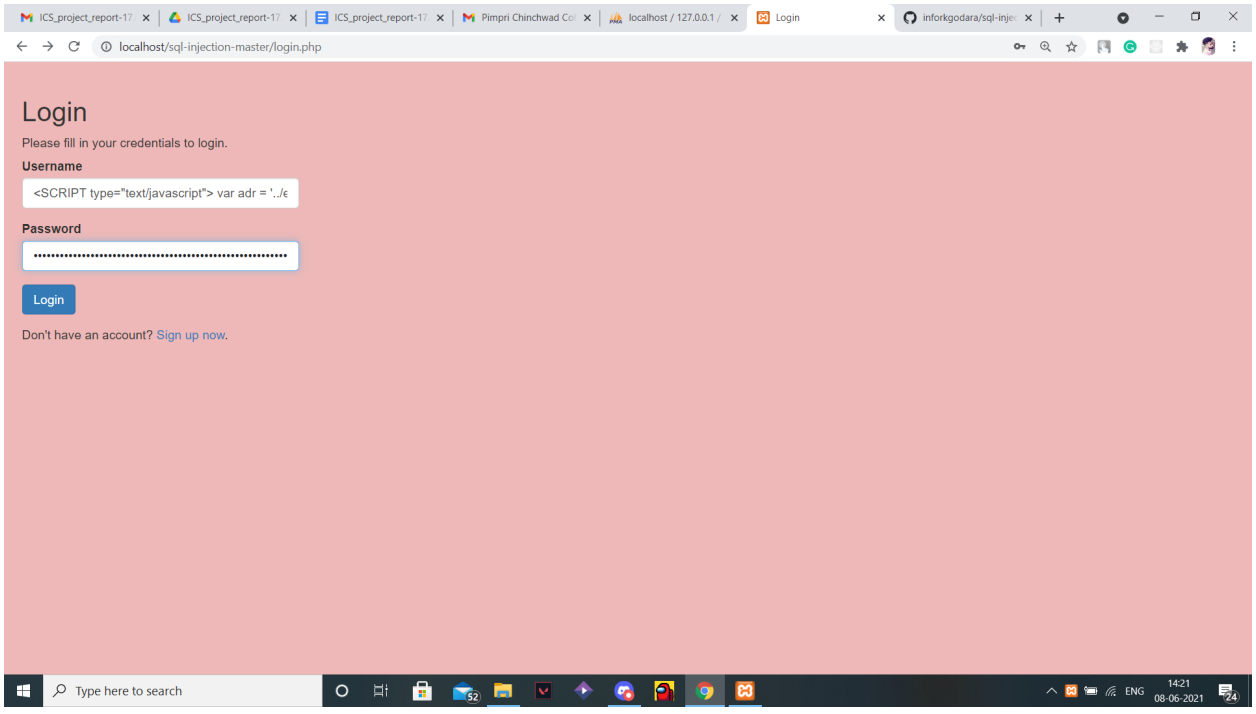

Fig14: Admin Login using Cross Scripting

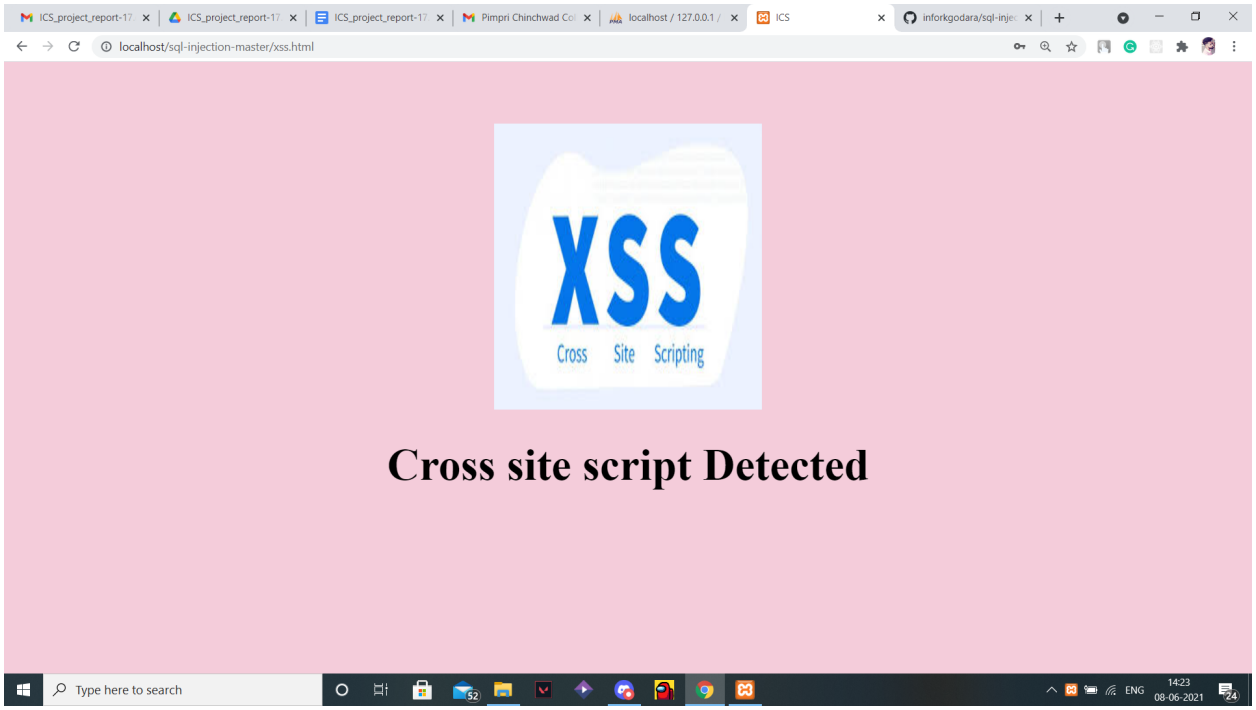

Fig15: Error Notification Of Cross Side Scripting.

7.CONCLUSION

The possibility of an SQL injection attack and Cross Scripting attack on web applications is high. The attacker can modify, delete data, perform database/server shutdown taking advantage of the vulnerabilities in the system. This paper presents the various attack methods, their classification using which the system administrators and programmers can understand SQLIA and secure the web application. However, as technology develops, so will the threats and techniques used by malicious users. As storage on the internet is more of a trend nowadays care should be taken to secure the data from being stolen by malicious users. Hence securing the system against SQL injection attack is of great importance

8.REFERENCES

[1] J. V. William G.J. Halfond and A. Orso, ―A classification of sql injection attacks and countermeasures,‖ 2006.

[2] A. Tajpour; M. Masrom; M. Z. Heydari.; S. Ibrahim; "SQL injection detection and prevention tools assessment, " Proc. Of ICCSIT 2010, vol.9, no., pp.518-522, 9-11 July 2010.

[3] IndraniBalasundaram. , Dr. E. Ramara. An Approach to Detect and Prevent SQL Injection Attacks in Database Using Web Service. IJCSNS International Journal of Computer Science and Network Security, VOL.11 No.1, YEAR 2012

[4] C. Anley. Advanced SQL Injection In SQL Server Applications. White paper, Next Generation Security Software Ltd., 2002.

[5] S. Labs. SQL Injection. White paper, SPI Dynamics, Inc.,

[6] Abhishek Kumar Baranwal. Approaches to detect SQL injection and XSS in web applications. EECE 571B, TERM SURVEY PAPER, APRIL 2012

[7] NehaSingh,Ravindra Kumar Purwar,SQL Injection –A HazardTo web applications, International Journal of Advanced Research in computer Science and Software Engineering,vol.2,Issue 6,June 2012,pp. 42-46.

[8] AnyiLiu , Yi Yuan , DumindaWijesekera , AngelosStavrou,SQLProb: A Proxy-based Architecture towards Preventing SQL Injection Attacks

[9] AtefehTajpour , SuhaimiIbrahim,MohammadSharifi,Web Application Security by SQL Injection Detection Tools,IJCSI,International Journal Computer Science Issues,Vol.9,Issue 2,No.3,March 2012,332-339

[10] StephenW.Boyd,AngelosD.Keromyti,SQLrand:Preventing SQL Injection Attacks.