Terna Engineering College

Computer Engineering Department

Class: TE **Sem.: VI**

Course: System Security Lab

PART A

(PART A : TO BE REFFERED BY STUDENTS)

Experiment No.14

A.1 Aim: Perform SQL injection on a vulnerable website.

A.2 Prerequisite:

1. Basic Knowledge of SQL queries, html/PHP.

A.3 Outcome:

After successful completion of this experiment students will be able to

To be able to set up firewalls and intrusion detection systems using open source technologies and to explore email security and explore various attacks like buffer-overflow, SQL injection and web-application attacks

A.4 Theory:

SQL Injection (SQLi) is a type of an injection attack that makes it possible to execute malicious SQL statements. These statements control a database server behind a web application. Attackers can use SQL Injection vulnerabilities to bypass application security measures. They can go around authentication and authorization of a web page or web application and retrieve the content of the entire SQL database. They can also use SQL Injection to add, modify, and delete records in the database.

An SQL Injection vulnerability may affect any website or web application that uses an SQL database such as MySQL, Oracle, SQL Server, or others. Criminals may use it to gain unauthorized access to your sensitive data: customer information, personal data, trade secrets, intellectual property, and more. SQL Injection attacks are one of the oldest, most prevalent, and most dangerous web application vulnerabilities.

To make an SQL Injection attack, an attacker must first find vulnerable user inputs within the web page or web application. A web page or web application that has an SQL Injection vulnerability uses such user input directly in an SQL query. The attacker can create input content. Such content is often called a malicious payload and is the key part of the attack. After the attacker sends this content, malicious SQL commands are executed in the database.

SQL is a query language that was designed to manage data stored in relational databases. You can use it to access, modify, and delete data. Many web applications and websites store all the data in SQL databases. In some cases, you can also use SQL commands to run operating system commands. Therefore, a successful SQL Injection attack can have very serious consequences.

- Attackers can use SQL Injections to find the credentials of other users in the database. They can then impersonate these users. The impersonated user may be a database administrator with all database privileges.
- SQL lets you select and output data from the database. An SQL Injection vulnerability could allow the attacker to gain complete access to all data in a database server.
- SQL also lets you alter data in a database and add new data. For example, in a financial application, an attacker could use SQL Injection to alter balances, void transactions, or transfer money to their account.
- You can use SQL to delete records from a database, even drop tables. Even if the administrator makes database backups, deletion of data could affect application availability until the database is restored. Also, backups may not cover the most recent data.
- In some database servers, you can access the operating system using the database server. This may be intentional or accidental. In such case, an attacker could use an SQL Injection as the initial vector and then attack the internal network behind a firewall.

PART B

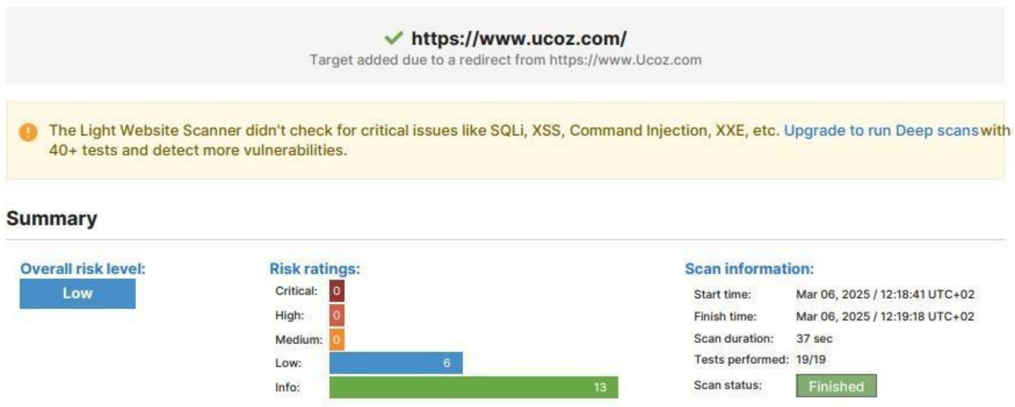
(PART B : TO BE COMPLETED BY STUDENTS)

(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case the there is no Black board access available)

Roll No. 30	**Name: Bhatt Pranjal Deepak**
Class : TE-B	**Batch :B2**
Date of Experiment:	**Date of Submission**
Grade :	

B.1 Output

ucoz and Facebook are slightly vulnerable, as they have a low level of security risk.



✓ https://www.ucoz.com/
Target added due to a redirect from https://www.Ucoz.com

The Light Website Scanner didn't check for critical issues like SQLi, XSS, Command Injection, XXE, etc. Upgrade to run Deep scans with 40+ tests and detect more vulnerabilities.

Summary

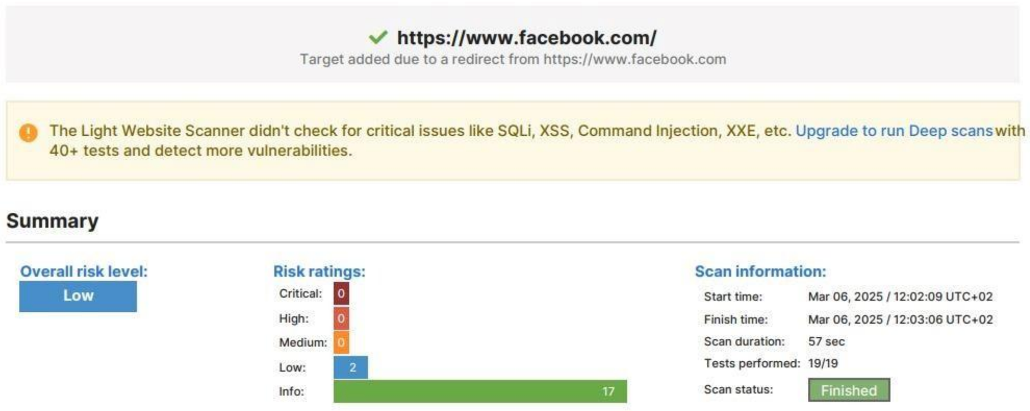
Overall risk level: Low

Risk ratings:

Critical:	0
High:	0
Medium:	0
Low:	6
Info:	13

Scan information:

Start time:	Mar 06, 2025 / 12:18:41 UTC+02
Finish time:	Mar 06, 2025 / 12:19:18 UTC+02
Scan duration:	37 sec
Tests performed:	19/19
Scan status:	Finished



✓ https://www.facebook.com/
Target added due to a redirect from https://www.facebook.com

The Light Website Scanner didn't check for critical issues like SQLi, XSS, Command Injection, XXE, etc. Upgrade to run Deep scans with 40+ tests and detect more vulnerabilities.

Summary

Overall risk level: Low

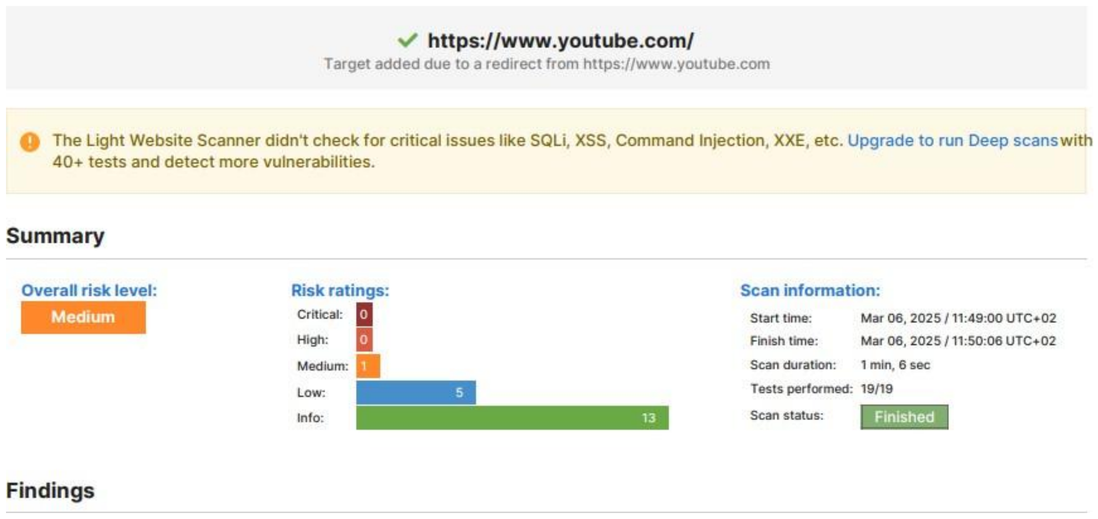
Risk ratings:

Critical:	0
High:	0
Medium:	0
Low:	2
Info:	17

Scan information:

Start time:	Mar 06, 2025 / 12:02:09 UTC+02
Finish time:	Mar 06, 2025 / 12:03:06 UTC+02
Scan duration:	57 sec
Tests performed:	19/19
Scan status:	Finished

YouTube and Instagram have a medium level of risk, indicating moderate security vulnerabilities



✓ **https://www.youtube.com/**

Target added due to a redirect from https://www.youtube.com

The Light Website Scanner didn't check for critical issues like SQLi, XSS, Command Injection, XXE, etc. Upgrade to run Deep scans with 40+ tests and detect more vulnerabilities.

Summary

Overall risk level: Medium

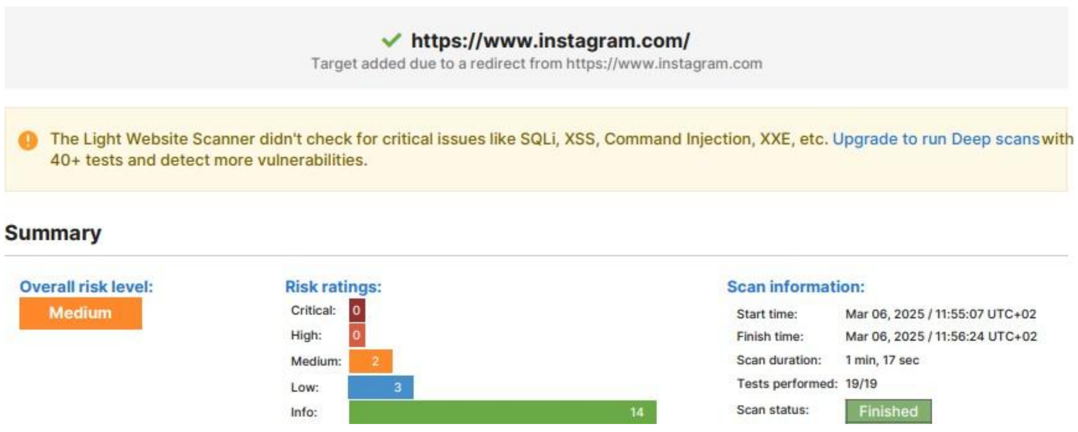
Risk ratings:

Critical:	0
High:	0
Medium:	1
Low:	5
Info:	13

Scan information:

Start time:	Mar 06, 2025 / 11:49:00 UTC+02
Finish time:	Mar 06, 2025 / 11:50:06 UTC+02
Scan duration:	1 min, 6 sec
Tests performed:	19/19
Scan status:	Finished

Findings



✓ **https://www.instagram.com/**

Target added due to a redirect from https://www.instagram.com

The Light Website Scanner didn't check for critical issues like SQLi, XSS, Command Injection, XXE, etc. Upgrade to run Deep scans with 40+ tests and detect more vulnerabilities.

Summary

Overall risk level: Medium

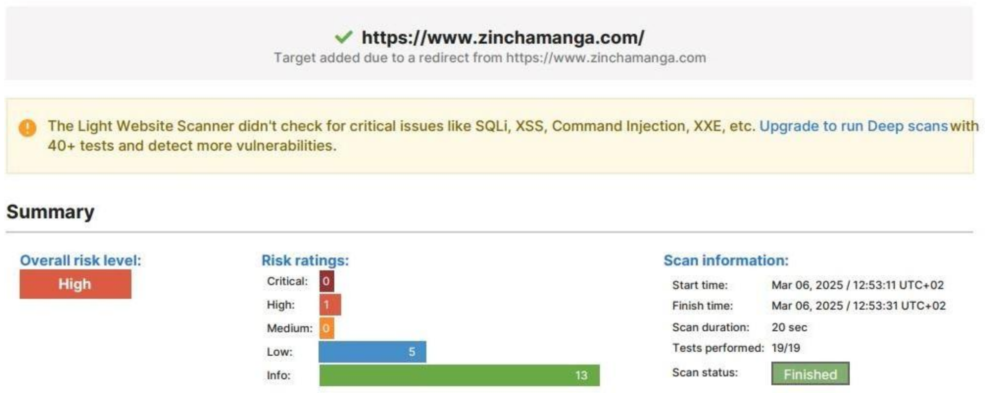
Risk ratings:

Critical:	0
High:	0
Medium:	2
Low:	3
Info:	14

Scan information:

Start time:	Mar 06, 2025 / 11:55:07 UTC+02
Finish time:	Mar 06, 2025 / 11:56:24 UTC+02
Scan duration:	1 min, 17 sec
Tests performed:	19/19
Scan status:	Finished

Zinchamanga and Cricbuzz are highly vulnerable due to their high level of security risk



✓ https://www.zinchamanga.com/
Target added due to a redirect from https://www.zinchamanga.com

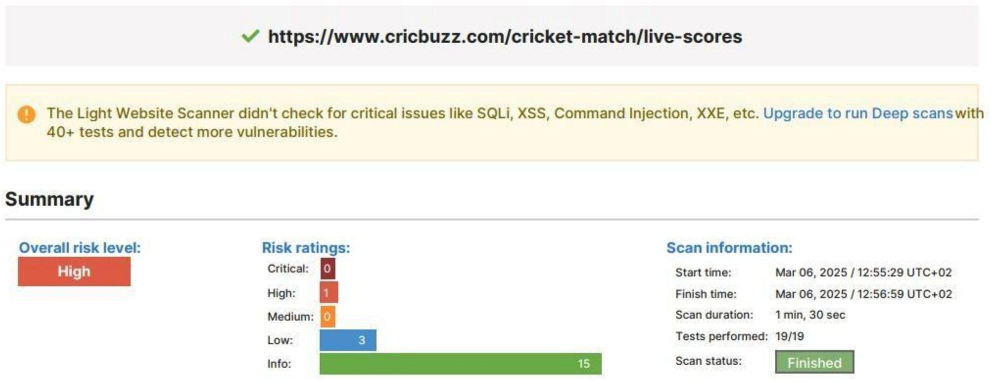
The Light Website Scanner didn't check for critical issues like SQLi, XSS, Command Injection, XXE, etc. Upgrade to run Deep scans with 40+ tests and detect more vulnerabilities.

Summary

Overall risk level: High

Risk ratings:	
Critical:	0
High:	1
Medium:	0
Low:	5
Info:	13

Scan information:	
Start time:	Mar 06, 2025 / 12:53:11 UTC+02
Finish time:	Mar 06, 2025 / 12:53:31 UTC+02
Scan duration:	20 sec
Tests performed:	19/19
Scan status:	Finished



✓ https://www.cricbuzz.com/cricket-match/live-scores

The Light Website Scanner didn't check for critical issues like SQLi, XSS, Command Injection, XXE, etc. Upgrade to run Deep scans with 40+ tests and detect more vulnerabilities.

Summary

Overall risk level: High

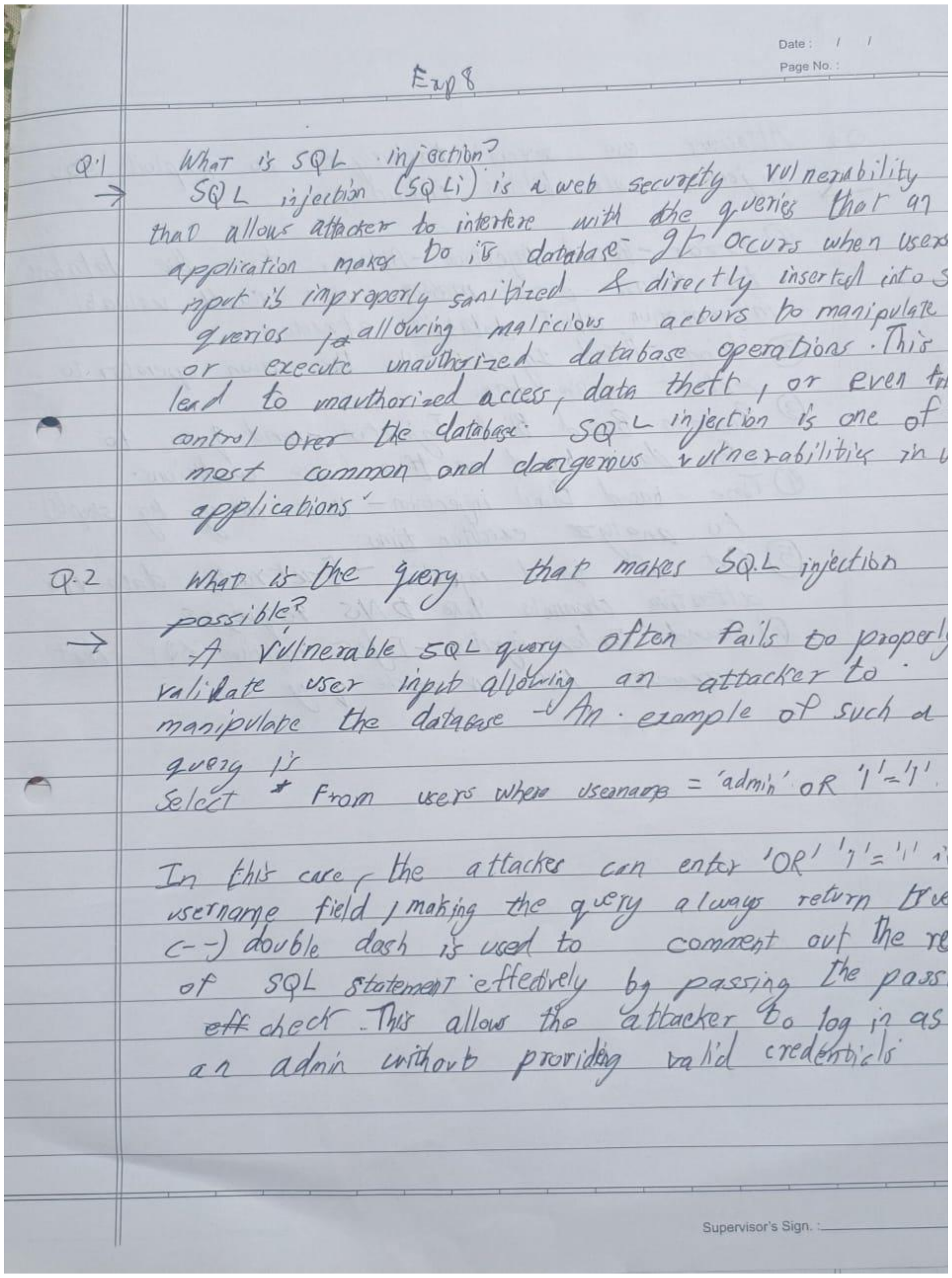
Risk ratings:	
Critical:	0
High:	1
Medium:	0
Low:	3
Info:	15

Scan information:	
Start time:	Mar 06, 2025 / 12:55:29 UTC+02
Finish time:	Mar 06, 2025 / 12:56:59 UTC+02
Scan duration:	1 min, 30 sec
Tests performed:	19/19
Scan status:	Finished

B.2. Commands / tools used with syntax: PENTEST

B.3 Question of Curiosity: *(Attempt at least 3 questions handwritten)*

1. What is SQL injection?



Date : / /
Page No. :

Exp 8

Q·1 What is SQL injection?
→ SQL injection (SQLi) is a web security vulnerability that allows attacker to interfere with the queries that an application makes to its database. It occurs when users input is improperly sanitized & directly inserted into S queries ~~to~~ allowing malicious actors to manipulate or execute unauthorized database operations. This lead to unauthorized access, data theft, or even to control over the database. SQL injection is one of most common and dangerous vulnerabilities in applications.

Q.2 What is the query that makes SQL injection possible?
→ A vulnerable SQL query often fails to properly validate user input allowing an attacker to manipulate the database. An example of such a query is
Select * From users where usernam = 'admin' OR '1'='1'.

In this care, the attacker can enter 'OR' '1'='1' i username field, making the query always return true (--) double dash is used to comment out the re of SQL statement effectively by passing the pass ~~eff~~ check. This allow the attacker to log in as an admin without providing valid credentials

Supervisor's Sign. :

3. Explain about “ OR 1 = 1” and what happens to SQL when this condition is used in the query.

   The "OR 1=1" condition is a common trick used in SQL Injection attacks. It exploits SQL queries by always evaluating as true, causing unintended behavior in the database.

- Consider this query:

SELECT * FROM users WHERE username = '' OR 1=1;

- Since 1=1 is always true, the query retrieves all users from the database instead of just a specific one.
- If this is an authentication query, it allows an attacker to log in as any user without knowing their credentials.
- If applied to DELETE or UPDATE queries, it can result in mass deletion or unwanted modification of records.

4. What are the potential risks and consequences of a successful SQL injection attack?

SQL Injection attacks pose serious risks to web applications and their users. Some of the potential consequences include:

- **Unauthorized Data Access** – Attackers can retrieve sensitive information such as usernames, passwords, credit card details, and personal records.
- **Data Manipulation** – Hackers can modify existing records, change permissions, or corrupt important data, affecting the integrity of the system.
- **Data Deletion** – Malicious queries can delete entire tables or databases, leading to permanent data loss.
- **Authentication Bypass** – Attackers can log in as any user, including administrators, without valid credentials.
- **Website Defacement** – Hackers can alter website content, post malicious messages, or redirect users to fraudulent sites.
- **Full Database Control** – Advanced SQL Injection techniques can allow an attacker to gain administrative control over the database, compromising the entire system.
- **Financial Loss and Reputation Damage** – Data breaches can lead to financial penalties, loss of customer trust, and legal consequences.

5. How can parameterized queries and prepared statements help prevent SQL injection?

One of the most effective ways to prevent SQL Injection is by using **parameterized**

queries and **prepared statements**. These techniques ensure that user inputs are treated strictly as data and not as executable SQL code.

- **Parameterized Queries**: These queries separate SQL commands from user input. Instead of directly inserting values into SQL strings, placeholders (? or :param) are used:

```
sql
SELECT * FROM users WHERE username = ? AND password = ?
```

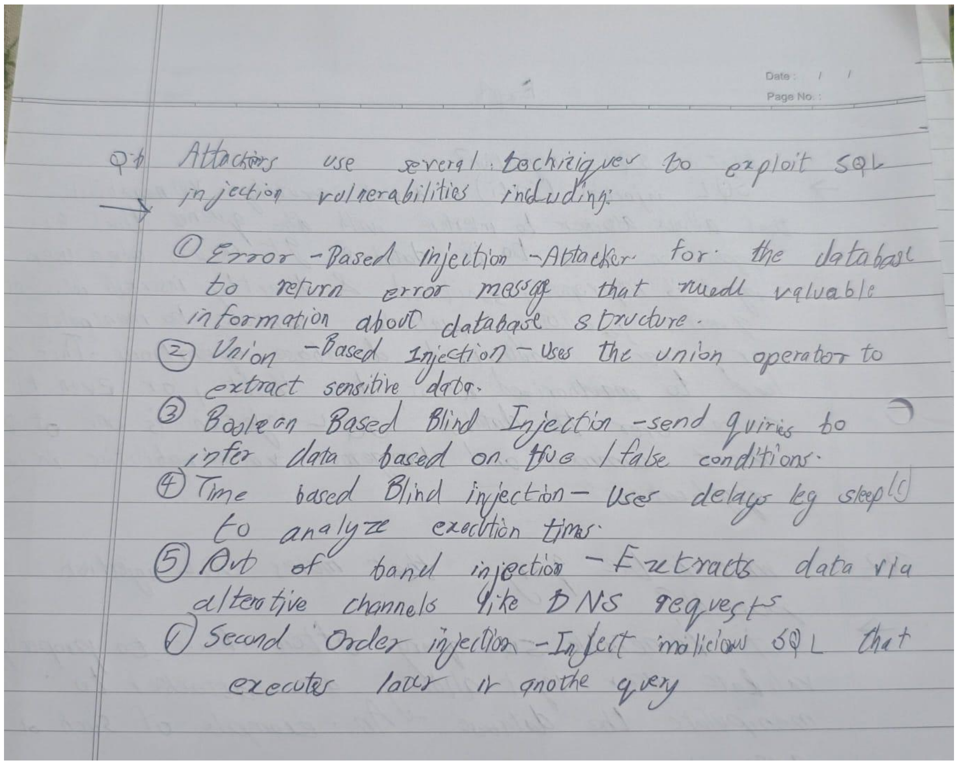
The actual values are safely bound to these placeholders, preventing attackers from injecting malicious SQL code.

- **Prepared Statements**: These are precompiled SQL statements that execute with user-provided values at runtime. They ensure that inputs are always treated as data and never executed as part of the query. Example in Python (using MySQL):

```
python
cursor.execute("SELECT * FROM users WHERE username = %s AND password = %s",
(user_input, password))
```

- **Benefits**:
    - Prevents attackers from modifying the structure of the SQL query.
    - Eliminates the risk of injecting additional SQL commands.
    - Ensures data integrity and security

6. What are some common methods attackers use to exploit SQL injection vulnerabilities?



Date : / /
Page No. :

Q.b Attackers use several techiqver to exploit SQL injection vulnerabilities including:

→

① Error - Based injection - Attacker for the database to return error message that ruedl valuable information about database structure.

② Union - Based Injection - Uses the union operator to extract sensitive data.

③ Boolean Based Blind Injection - send quiries to infer data based on true / false conditions.

④ Time based Blind injection - Uses delays leg sleep() to analyze execution times.

⑤ Out of band injection - Extracts data via alterative channels like DNS requests

⑥ Second Order injection - Infect malicious SQL that executes later ir anothe query

B.4 Conclusion:

(Write appropriate conclusion.)

Conclusion

The SQL injection vulnerability assessment conducted using a penetration testing tool revealed varying levels of security across different websites. Zinchamanga and Cricbuzz were found to have high-level risks, making them more susceptible to potential attacks that could compromise user data and system integrity. YouTube and Instagram exhibited medium-level risks, indicating partial security measures that might still leave certain vulnerabilities open to exploitation. In contrast, Ucoz and Facebook showed low-risk levels, suggesting better security defenses against SQL injection attacks.

These findings highlight the critical need for robust database security practices, such as implementing prepared statements, parameterized queries, and strict input validation. Regular security assessments and proactive measures can help minimize vulnerabilities, ensuring better protection against cyber threats.