

Forum:[Hack Forums](#)

Thread: [\[CYBERSEC SERIES #7/10 \] SQL Injection Explained How Websites Get Exploited](#)

Mastercarder - May 27, 2026, 2:22 pm

[CYBERSEC SERIES #7/10]

SQL Injection Explained — How Websites Get Exploited

Educational purposes only. Only test applications you own or have permission to assess.

What Is SQL Injection?

SQL Injection, also known as SQLi, is one of the oldest yet still dangerous web vulnerabilities.

It happens when a website does not properly handle user input before sending it into a database query.

When protections are missing, an attacker may be able to manipulate the query and make the application behave in unintended ways.

This can potentially allow:

- login bypasses
- sensitive data exposure
- database manipulation
- account compromise
- unauthorized access to web applications

1 — What Is SQL?

SQL stands for Structured Query Language.

Websites commonly use SQL databases to store:

- usernames
- passwords
- emails
- posts
- payment information
- application data

When a user logs into a website, the server may run a query similar to this:

Code `SELECT * FROM users`

```
WHERE username = 'admin'  
AND password = 'password123';  
If a matching account exists, access is granted.
```

2 — How SQL Injection Happens

The vulnerability appears when applications directly trust user input.

Example of unsafe PHP code:

```
Code $query = "SELECT * FROM users  
WHERE username='$username'  
AND password='$password'";
```

Because user input is inserted directly into the query, improperly secured applications may accidentally allow attackers to alter the SQL logic itself.

That is what makes SQL Injection so dangerous.

3 — SQL Injection Logic

Basic attack flow:

1. User enters crafted input
2. Application processes it incorrectly
3. SQL query structure changes
4. Database executes the modified query
5. Unauthorized access or data exposure may occur

This is why secure coding practices are critical in web development.

4 — Why It Was So Dangerous

Older websites often lacked:

- proper input sanitization
- prepared statements
- secure frameworks
- parameterized queries

Because of this, attackers could sometimes:

- bypass authentication systems
- retrieve database information
- manipulate records
- extract sensitive user data

Even major companies have suffered breaches caused by SQL Injection vulnerabilities.

5 — Preventing SQL Injection

Modern developers defend against SQL Injection using:

- prepared statements
- parameterized queries
- strict input validation
- ORM frameworks
- least-privilege database permissions

Safer example using PHP PDO:

```
Code $stmt = $pdo->prepare(
"SELECT * FROM users
WHERE username = ?
AND password = ?"
);
```

```
$stmt->execute([$username, $password]);
```

This separates user input from executable SQL commands.

6 — Practice Safely & Legally

Never test SQL Injection against real websites without authorization.

If you want to learn ethically, use legal lab environments such as:

- DVWA
- OWASP Juice Shop
- TryHackMe
- Hack The Box

These platforms are intentionally designed for cybersecurity training and education.

Final Thoughts

SQL Injection is a perfect example of how small coding mistakes can create massive security risks.

Understanding SQLi helps developers build safer applications and helps cybersecurity learners understand how modern web security evolved.

Series #8 coming soon.

MasterCard @ HF