Preventing SQL Injection Attacks in Web Application Development

- Introduction to SQL Injection
- Problem Description
- Prevention
- Demo
- Conclusion

SQL Injection

SQL Injection is a cyber-attack that targets web applications that rely on SQL databases.

It can affect anyone who uses any web or mobile application that involves databases.

Attackers can exploit vulnerabilities and inject malicious SQL statements to view, create, delete, or modify databases.

Main Consequences

Confidentiality: Loss of confidentiality is a frequent problem with SQL Injection vulnerabilities.

Authentication: If poor SQL commands are used to check usernames and passwords, it may be possible to connect to a system as another user with no previous knowledge of the password.

Authorization: If authorization information is held in a SQL database, it may be possible to change this information by successfully exploiting the SQL Injection vulnerability.

Integrity: Just as it may be possible to read sensitive information, it is also possible to make changes or delete it with a SQL Injection attack.

Types of SQL Injection

1. In-band SQL Injection
   - Error-based SQL injection
   - Union-based SQL injection
2. Inferential SQL Injection
   - Boolean injection
   - Time-based injection
3. Out-of-Band SQL Injection

SQL Injection

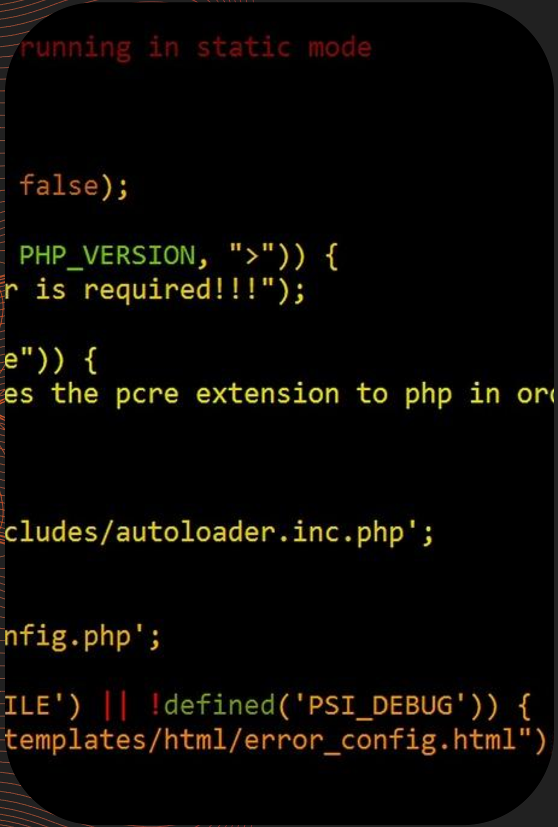
WHERE IT HAPPENS? Typically server-side of web or mobile applications

WHEN IT HAPPENS? It can happen anytime!

WHY WE NEED TO SECURE? Prevent legal issues, financial losses, data breaches, etc

HOW IT HAPPENS? Improper or insufficient security

How: Illustration of SQL injection



```
running in static mode



 false);

 PHP_VERSION, ">")) {
r is required!!!");

e")) {
es the pcre extension to php in ord



cludes/autoloader.inc.php';



nfig.php';

ILE') || !defined('PSI_DEBUG')) {
templates/html/error_config.html")
```

Demo: SQL Injection Attack

With a PHP login page without SQL Injection prevention, we'll login with malicious credentials which are an SQL statement.

Demo: SQL Injection Prevention

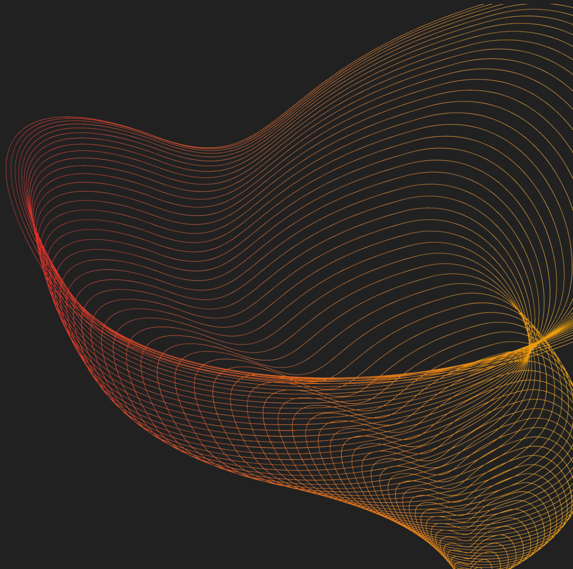
Now, we have implemented SQL Injection prevention using prepared statements. Let's try the malicious credentials again and see the difference.

Risks need to be mitigated

- **Risk of not using prepared statements**
- Insecure use of prepared statements
- Plaintext password
- Password complexity and account lockouts
- Other forms of injection attacks

Conclusion: SQL Injection Prevention

This demonstration underscores the need to prevent SQL Injection attacks in our web applications for data security and system integrity.



Work Plan

Component/Deliverable	Hours per Person (Total)	Group Member
Introduction How: Illustration of SQL Injection Vulnerability in a PHP Login Page Solution Description and Results Demo	approx 4 hours	Daniel
Problem Description ● Who: Who cares and who does it affect ● What: What is it ● How: How to detect SQL injection vulnerabilities ● Where: Where the problem is—client- or server-side ● When: Timing aspects ● Why: Laws, regulations, and other constraints Work Plan	approx 4 hours	Diana
The Main Consequences Types of SQL Injection What is the impact of a successful SQL Injection attack	approx 4 hours	Soha
Lessons Learned Conclusion Video Editing	approx 4 hours	Zixuan

Q & A

😊