PROJECT PROPOSAL

SQL Injection Attack: Advanced Methodologies and Defense

Problem Statement:

SQL injection attack is a critical security vulnerability that allows attackers to manipulate and access database contents by injecting malicious SQL statements through input fields. This can lead to unauthorized data access, data corruption, and severe breaches in data-driven applications. Despite being a well-known issue, SQL injection attacks remain prevalent due to inadequate input validation and insufficient use of prepared statements. This project aims to explore advanced SQL injection technique, focusing on Tautology while also presenting robust defense mechanism.

Project Objective:

The primary objective of this project is to demonstrate advanced SQL injection attack methodologies and provide comprehensive defense strategies to mitigate these vulnerabilities. Using a PHP web application powered by Apache, MariaDB, and phpMyAdmin, we will simulate SQL injection attack, focusing on Tautology and showcase effective countermeasure, emphasizing the importance of secure coding practices and the use of prepared statements and parameterized queries.

Stakeholders:

- **Developers:** Responsible for the design, implementation, and demonstration of SQL injection attacks and defenses.
- **Educational Institutions:** Potential stakeholders interested in using the project for educational purposes to teach cyber attack and defense concepts.
- **End Users:** Benefit indirectly through improved security of web applications.

Project Deliverables:

1. **MS PowerPoint Presentation:** Comprehensive PPT slide on SQL injection attack and defense.
2. **Vulnerable PHP Web Application:** A deliberately vulnerable web application built using Apache, PHP, MariaDB, and phpMyAdmin to demonstrate SQL injection attack.
3. **Attack Demonstration:** Detailed demonstrations of advanced SQL injection attack techniques on the vulnerable web application.
4. **Defense Mechanism:** Implementation and documentation of effective defense, like prepared statements, parameterized queries, and the use of secure PHP frameworks.
5. **Final Report:** A comprehensive report summarizing the project, including methodology, assessment, and recommendations for preventing SQL injection attacks.