SQL Injection Attacks: Mechanisms, Risks, and Mitigation Techniques

Author: Alex Mbogo
Email: evansnjiru5@gmail.com
Date: June 2025

Abstract

SQL Injection (SQLi) is a well-documented web application security vulnerability that allows attackers to interfere with the queries an application makes to its database. This paper investigates the technical mechanisms behind SQL injection, highlights real-world impacts using the Sony PlayStation Network breach as a case study, and proposes best-practice mitigation strategies. A simplified diagram is presented to visualize how SQLi works within web applications.

Nb: This For education purposes only

1. Introduction

Structured Query Language (SQL) Injection remains a top web application security risk (OWASP, 2021). Through unsanitized inputs, attackers manipulate queries to exfiltrate or manipulate sensitive data. Despite awareness, many modern applications still lack effective defense mechanisms.

2. How SQL Injection Works

SQLi allows an attacker to inject malicious SQL statements into application input fields. This typically exploits poor input validation or improperly constructed queries.

2.1 Mechanism Flow Diagram

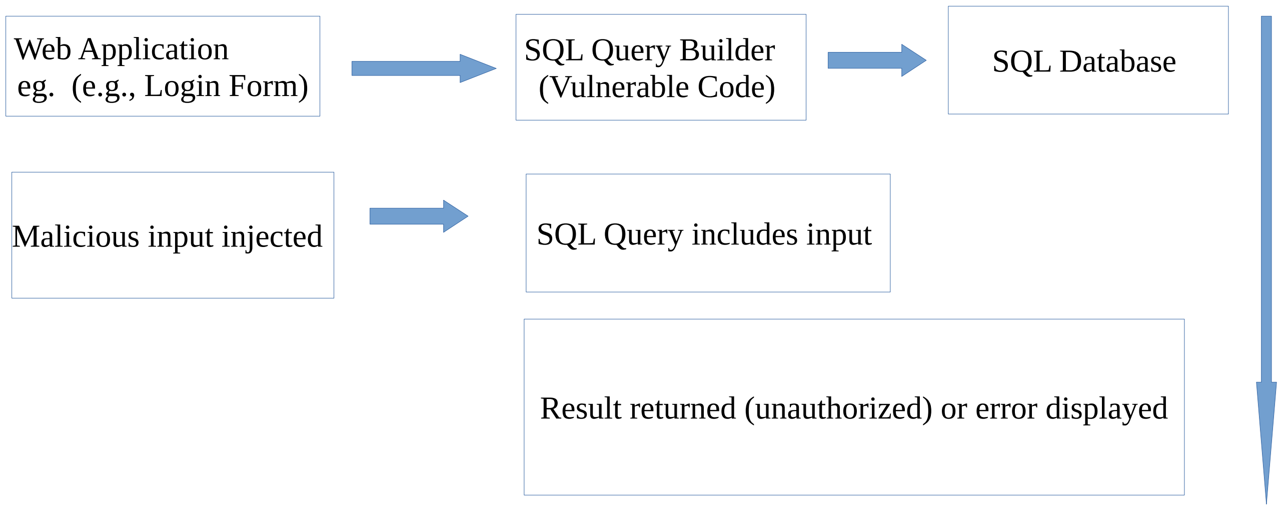

Figure 1: SQLi Flow – from user input to database compromise.

3. Common SQL Injection Techniques

3.1 Boolean-Based Blind SQLi

Returns different results depending on true/false conditions.

id=1 AND 1=1 -- true

id=1 AND 1=2 – false

3.2 Time-Based Blind SQLi

Detects information by measuring response time:

id=1 AND SLEEP(5)

3.3 Error-Based SQLi

Forces SQL errors that disclose internal DB structure.

id=1 AND (SELECT 1 FROM (SELECT COUNT(*), CONCAT(…)))

3.4 UNION-Based SQLi

Extracts data via a crafted `UNION SELECT` query.

id=1 UNION ALL SELECT NULL, NULL, username, password FROM users-- -

4. Case Study: The Sony PlayStation Network Breach (2011)

In 2011, Sony suffered a breach affecting over 77 million PlayStation Network (PSN) users. Reports indicated that SQL injection was one of the vulnerabilities used in the attack.

Attack Overview:

- Exploited poorly secured web application.
- Gained unauthorized access to Sony's customer database.
- Extracted usernames, passwords, emails, and purchase history.
- Sony estimated a cost of over **$171 million** in damages.

> *"Sony is deeply sorry for the inconvenience and concern caused by this attack."*
> — Sony Press Statement, 2011

This incident highlighted the massive financial and reputational risks associated with SQLi vulnerabilities.

---

5. Mitigation Strategies

5.1 Parameterized Queries

Ensures inputs are treated as data, not executable code:

cursor.execute("SELECT * FROM users WHERE username = ?", (input,))

5.2 Input Validation and Escaping

Use whitelisting to restrict expected input formats.

5.3 Use of ORM Libraries

Frameworks like Sequelize, Django ORM, or Hibernate abstract raw SQL queries.

5.4 Web Application Firewalls (WAF)

Detect and block malicious requests in real time.

5.5 Regular Security Testing

Employ tools like:

- **SQLMap** for penetration testing.
- **Burp Suite** for intercepting and modifying requests.
- **Static analysis tools** (e.g., SonarQube) for code auditing.

---

6. Conclusion

SQL injection is a dangerous yet preventable vulnerability. As seen in the Sony breach, its consequences can be catastrophic. By understanding its mechanisms and implementing layered defenses—from secure coding to runtime protections—developers can drastically reduce the risk of exploitation.

References

- Buehrer, G. T., Weide, B. W., & Sivilotti, P. A. G. (2005). *Using Parse Tree Validation to Prevent SQL Injection Attacks*. International Workshop on Software Engineering and Middleware.
- Halfond, W. G. J., Viegas, J., & Orso, A. (2006). *A Classification of SQL Injection Attacks and Countermeasures*. Proceedings of the IEEE International Symposium on Secure Software Engineering.
- OWASP. (2021). *OWASP Top 10: 2021*. https://owasp.org/Top10/
- Poulsen, K. (2009). *Heartland Payment Systems Hack Resulted in Loss of 100M+ Card Numbers*. Wired. https://www.wired.com/2009/01/heartland/
- Sony Press Release. (2011). *Cyberattack Statement*. https://www.sony.com/

Nb: This For education purposes only