**CTF Challenge:SQL INJECTION**

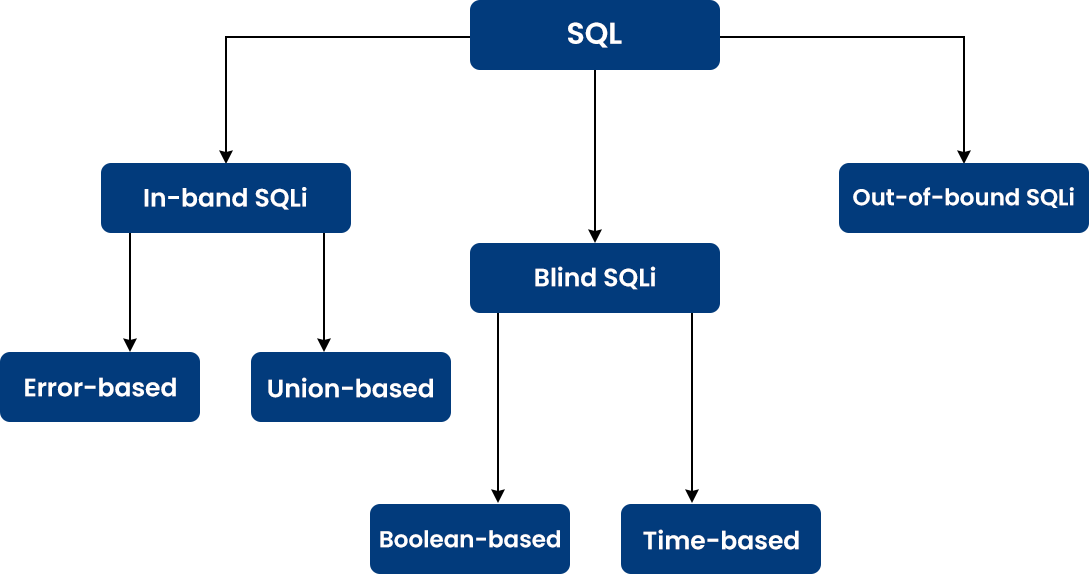
****

**SQL Injection?**

****

"SQL Injection" (SQLi) is a web security vulnerability that allows an attacker to interfere with the queries an application makes to its database. It generally allows attackers to view data that they are not normally able to retrieve. In many cases, an attacker can modify or delete this data, causing persistent changes to the application's content or behavior. In some cases, SQL injection can also be used to execute administrative operations on the database, recover the contents of files, and in certain situations, issue commands to the operating system.

### **Types of SQL Injections**



1. **In-band SQLi (Classic SQLi)**:
   * **Error-based SQLi**: Relies on error messages thrown by the database server to obtain information about the structure of the database.
   * **Union-based SQLi**: Uses the UNION SQL operator to combine the results of two or more SELECT statements into a single result set, which is then returned as part of the HTTP response.
2. **Inferential SQLi (Blind SQLi)**:
   * **Boolean-based Blind SQLi**: Sends a SQL query to the database, which forces the application to return a different result depending on whether the query returns true or false.
   * **Time-based Blind SQLi**: Sends a SQL query to the database, which forces the database to wait (for a period of time) before responding, indicating whether the query returned true or false.
3. **Out-of-band SQLi**: Relies on the database server's capability to make DNS or HTTP requests to deliver data to an attacker. This type of attack is less common due to the reliance on specific functionalities.

### 

### **Common SQL Injection Techniques**

* **Tautology**: Injecting a statement that always evaluates to true, allowing bypass of authentication.
* **Union Queries**: Exploiting the UNION operator to combine results from multiple queries.
* **Piggybacked Queries**: Adding additional queries to the original one to execute multiple commands.
* **Stored Procedures**: Exploiting vulnerabilities in stored procedures or functions.
* **Inference**: Using conditional responses or time delays to infer information.

# **Capture the Flag (CTF) Challenges**

**Flag 1. Which SQL injection type relies on error messages thrown by the database server to obtain information?**

Answer: Error-based

**Flag 2. What technique uses the UNION SQL operator to combine the results of multiple SELECT statements?**

Answer: Union-based

**Flag 3. Which open-source tool automates the process of detecting and exploiting SQL injection flaws?**

Answer: SQLMap

**Flag 4. What method involves forcing the database to wait before responding to infer information about the query result?**

Answer: Time-based Blind SQLi

**Flag 5. What can be used to ensure that input data is treated as data and not executable code in SQL queries?**

Answer: Prepared Statements