SQL INJECTION

WHAT IS SQL?

SQL is Structured Query Language, which is a computer language for storing, manipulating and retrieving data stored in a relational database.

SQL is the standard language for Relational Database System. All the Relational Database Management Systems (RDMS) like MySQL, MS Access, Oracle, Sybase, Informix, Postgres and SQL Server use SQL as their standard database language.

EXAMPLE

A database models real-life entities like professors and universities by storing them in tables. Each table contains data from a single entity type. This reduces redundancy by storing entities only once. there only needs to be one row of data containing a certain company's details.

SOME OF THE MOST COMMON SQL COMMANDS

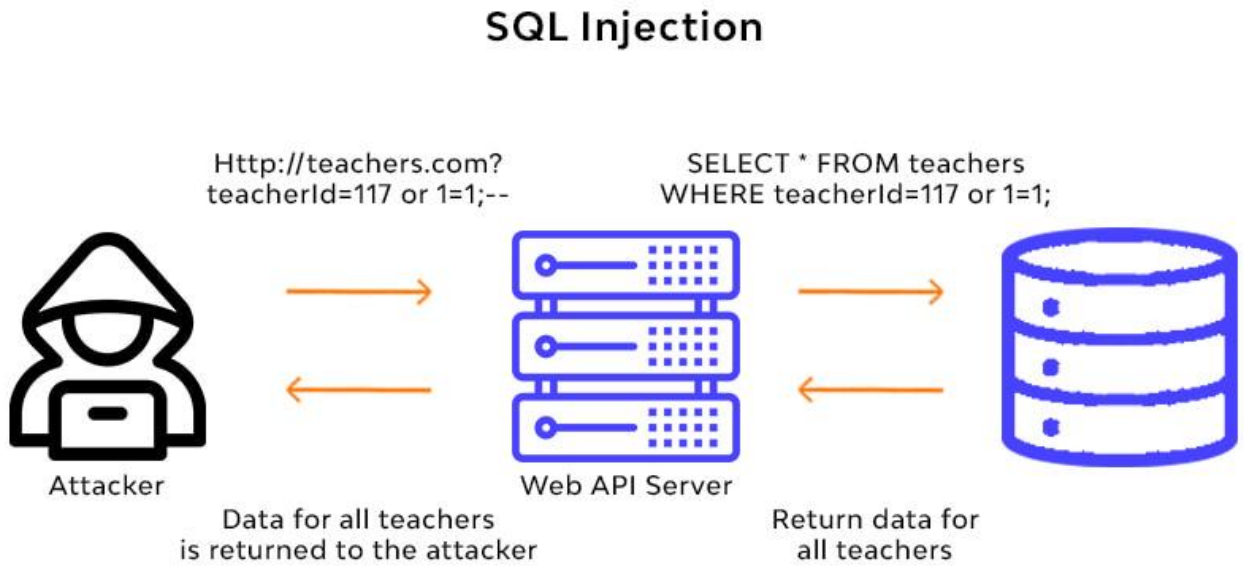
- SELECT - extracts data from a database
- UPDATE - updates data in a database
- DELETE - deletes data from a database
- INSERT INTO - inserts new data into a database
- CREATE DATABASE - creates a new database
- ALTER DATABASE - modifies a database
- CREATE TABLE - creates a new table
- ALTER TABLE - modifies a table
- DROP TABLE - deletes a table
- CREATE INDEX - creates an index (search key)
- DROP INDEX - deletes an index

WHAT IS SQL INJECTION?

SQL injection is a code injection technique used to attack data-driven applications, in which malicious SQL statements are inserted into an entry field for execution to dump the database contents to the attacker.

SQL injection must exploit a security vulnerability in an application's software, for example, when user input is either incorrectly filtered for string literal escape characters embedded in SQL statements or user input is not strongly typed and unexpectedly executed. SQL injection is mostly known as an attack vector for websites but can be used to attack any type of SQL database.

SQL injection attacks allow attackers to spoof identity, tamper with existing data, cause repudiation issues such as voiding transactions or changing balances, allow the complete disclosure of all data on the system, destroy the data or make it otherwise unavailable, and become administrators of the database server.



TYPES OF SQL INJECTION

SQL Injection can be classified into three major categories

1.In-band SQL injection (Classic SQL injection)

- In-band SQL Injection is the most common and easy-to-exploit of SQL Injection attacks.
- In-band SQL Injection occurs when an attacker is able to use the same communication channel to both launch the attack and gather results.

The two most common types of in-band SQL Injection are

Error-based SQL injection

- Error-based SQL injection SQL Injection technique that relies on error messages thrown by the database server to obtain information about the structure of the database.
- In some cases, error-based SQL injection alone is enough for an attacker to enumerate an entire database. While errors are very useful during the development phase of a web application, they should be disabled on a live site, or logged to a file with restricted access instead.

Union-based SQL injection

Union-based SQL injection is an SQL injection technique that leverages the UNION SQL operator to combine the results of two or more SELECT statements into a single result which is then returned as part of the HTTP response.

2.Inferential SQL injection (Blind SQL injection)

- IT may take longer for an attacker to exploit; however, it is just as dangerous as any other form of SQL Injection.
- In an inferential SQL injection attack, no data is actually transferred via the web application and the attacker would not

be able to see the result of an attack in-band (which is why such attacks are commonly referred to as “blind SQL Injection”).

- Instead, an attacker is able to reconstruct the database structure by sending payloads, observing the web application’s response and the resulting behaviour of the database server.

The two types of inferential SQL Injection are

Boolean-based (content-based) Blind SQL injection

- Boolean-based SQL Injection is an SQL Injection technique that relies on sending an SQL query to the database which forces the application to return a different result depending on whether the query returns a TRUE or FALSE result.
- Depending on the result, the content within the HTTP response will change, or remain the same. This allows an attacker to infer if the payload used returned true or false, even though no data from the database is returned. This attack is typically slow (especially on large databases) since an attacker would need to enumerate a database, character by character.

Time-based Blind SQL injection

- Time-based SQL Injection is an inferential SQL Injection technique that relies on sending an SQL query to the database which forces the database to wait for a specified amount of time (in seconds) before responding. The response time will indicate to the attacker whether the result of the query is TRUE or FALSE.
- Depending on the result, an HTTP response will be returned with a delay, or returned immediately. This allows an attacker to infer if the payload used returned true or false, even though no data from the database is returned. This attack is typically slow (especially on large databases) since an attacker would need to enumerate a database character by character.

3.Out-of-band SQL injection

- Out-of-band SQL Injection is not very common, mostly because it depends on features being enabled on the database server being used by the web application. Out-of-band SQL Injection occurs when an attacker is unable to use the same channel to launch the attack and gather results.
- Out-of-band techniques, offer an attacker an alternative to inferential time-based techniques, especially if the server responses are not very stable (making an inferential time-based attack unreliable).
- Out-of-band SQLi techniques would rely on the database server's ability to make DNS or HTTP requests to deliver data to an attacker. Such is the case with Microsoft SQL Server's xp_dirtree command, which can be used to make DNS requests to a server an attacker controls; as well as Oracle Database's UTL_HTTP package, which can be used to send HTTP requests from SQL and PL/SQL to a server an attacker controls.

WHAT IS IMPACT OF SQL INJECTION

SQL Injection are highly critical issues as these can be used to extract complete database contents and in some cases can be leveraged to a Command execution on the server

HOW TO PREVENT SQL INJECTIONS.

Step 1: Train and maintain awareness

To keep your web application safe, everyone involved in building the web application must be aware of the risks associated with SQL Injections. You should provide suitable security training to all your developers, QA staff, DevOps, and Sysadmins.

Step 2: Don't trust any user input

Treat all user input as untrusted. Any user input that is used in an SQL query introduces a risk of an SQL Injection. Treat input from authenticated and/or internal users the same way that you treat public input.

Step 3: Use whitelists, not blacklists

Don't filter user input based on blacklists. A clever attacker will almost always find a way to circumvent your blacklist. If possible, verify and filter user input using strict whitelists only.

Step 4: Adopt the latest technologies

Older web development technologies don't have SQL injection protection. Use the latest version of the development environment and language and the latest technologies associated with that environment/language. For example, in PHP use PDO instead of MySQL.

Step 5: Employ verified mechanisms

Don't try to build SQLi protection from scratch. Most modern development technologies can offer you mechanisms to protect against SQLi. Use such mechanisms instead of trying to reinvent the wheel. For example, use parameterized queries or stored procedures.

Step 6: Scan regularly

SQL Injections may be introduced by your developers or through external libraries/modules/software. You should regularly scan

your web applications using a web vulnerability scanner.