The attacker poisons dynamic SQL statements in the SQL Injection Attack to comment on some components of the declaration or to add a condition that will always be valid. The attacker uses the design faults to exploit SQL statements by implementing malicious SQL code is poorly designed web applications.

Usually, SQL injection happens when input is taken from a user, such as username feilds,idparameters etc the attacker will inject the sql statements which will directly executed by sql parser and attacker will tries to control a database server in order to retrieve data from database And also be used to bypass authentication.

Sql injection vulnerability can found in web applications that are using the sql databases like MySQL, Oracle, SQL Server, or others. SQLI is a common attack vector by this means the attacker can pass the malicious payloads to sql parsers in order to manipulation or access confidential , sensitive information like customer information, personal data etc that is stored in database.SQL Injection attack is one of the most common and dangerous Vulnerability.

An attacker must first discover vulnerable user inputs fields and parameters in the web application to perform a SQL Injection attack and such user input fields will be used to pass sqlinjection payload for performing SQL Injection attack. SQL injection also termed SQLI. Input content crafted by the attacker for injection is referred as a malicious payload and is the main component of the attack. After the attacker modifies the request parameters with malicious payloads then database executes malicious SQL commands and gives output with database content relevant to payload. The malicious queries can be inserted by the attacker via a web form or by attaching them directly to the end of the URL or HTTP headers.

SQL is a query language for managing data stored in relational databases. And it can be used to access, edit, and delete data. Many websites and web applications manage all the data in SQL databases. You can also use SQL commands to execute operating system commands in some instances. An effective SQL Injection attack can, therefore, have very severe implications like [2].

1. Attackers can use SQL Injection to identify other user's credentials in the database. These credentials can then be used for impersonated the other users . The impersonated user can be an administrator with all the privileges of the database also .
2. SQL allows you to select and display information in the database. An SQL Injection vulnerability could give the attacker full access to all information on a database server.
3. SQL also allows you to change information and add new information to a database. For instance, an attacker could use SQL Injection in a financial application to change balance, void transactions, or transfer cash to their account.
4. To delete documents from a database, you can use SQL, even to drop tables also. Even if database backups are made by the administrator, data deletion could influence the accessibility of the application until the database is restored. Backups may not also contain the latest information .
5. In some database servers, you can use the database server to access the working system. This may be accidental. In such a case, an attacker might use an SQL Injection to attack the internal network.

Types of SQL injection

**In-band SQLi**

The attacker utilizes the same communication channel to launch their attack and collect outcomes. The simplicity and effectiveness of In-band SQLI make it one of the most popular SQLi attack This technique has two sub-variations

Error-based SQLi— The attacker executes activities that cause error messages to be generated by the database. The attacker may use the data supplied by the error messages to collect information about the database structure [3].

Union-based SQLi—This method uses the UNION SQL operator to fuse various select statements generated by the database to obtain a single HTTP response. This result may include information that the attacker can leverage [3].

**Inferential (Blind) SQLi**

The attacker sends payloads to the data base server and observes the server's response and behavior in order to more information about database structure because in blind SQLI attacker cannot see or get much information in response. Blind SQL injections depend on the server's response and behavior patterns, to perform these types of attacks typically consume time because the attacker has to retrieve the character by character but can be just as damaging. The following can be types of blind SQL injection:

Boolean— the attacker will constructs Boolean based payloads and sends to the database through sql statements in such a way that the response from data base will change based on the weather condition is true of false. Based on the result, the HTTP response data will change or remain unchanged. by using this change in response based on condition the attacker can retrieve the database content

Time-based—

In time based SQL injection techniques the attacker will tries to inject payloads which will makes database to give response with time delay based on payload. Here the attacker can use SQL methods like SLEEP (), BENCHMARK (), WAITFORDELAY () etc which have specific special DBMS function (or) can use heavy query as payloads which will helps to generate the time delays. Based on time delay in the response the attacker can guess some kind of information.

**Out-of-band SQLi:**

Out - of-band SQLi will be used only if the attacker is unable to start the attack and retrieve the data using same channel, or if a server is too slow or unstable to perform such activities. in this method attacker will relay on servers ability to generate DNS or HTTP request which help to transfer the data gained from sql injection. The out of band sqlinjection will only possible if and only if the relevant features are enabled and have access to end-user on web application database server.

The best Preventive measures for SQL Injection Attacks are :

* Instead of placing user-supplied input directly into SQL statements, we use prepared statements with parameterized queries can be used or stored procedures can also be used which is harder to implement but much effective way to prevent sql injection.
* Sanitize user-provided inputs to properly escape the special characters and should verify that the input data is in predefined format or not.
* Don't leave sensitive data in plaintext while storing in the database instead Encrypt or hash the confidential data which will provides a further level of protection if the attacker successfully enters into the system.
* Restrict the rights and privileges of the database by reducing the user's capabilities to the bare minimum. This will restrict intruder or attacker if they succeed in gaining access.
* Displaying the common database error messages directly to end user should be avoided instead the custom error messages should be displayed for end user
* By using the Web Application Firewall (WAF) for web applications with database access which helps to protects web-based applications from malicious payload injection attempts by identifying and blocking malicious payloads based on firewall configuration rules.