

SQL injection is a method that takes advantages of how a server interact or works with a backend database server.

- In a SQL injection attack, a server is tricked in to passing SQL command through the web application then get processed and executed by the backend database server and some other benefits of successful attack could be Bypassing the authentication or Gaining access to data, changing data, execution of code etc...
- Types and method :- It depends on types of database server that is being used on target.

If you are working in a company, and/or you are responsible for analyze threats, one method to know if your application has been a target for the SQL injection, where some attackers tried to attack it; go to the log file of your server back-end side and try to find any logging attempts and/or errors.

- SQL Injection can be done even from browsing a site if it has database search option there. So URL can also indicate the actual query has been sent.
- There are three types of SQL Injection
  - Error Based
  - Union Based
  - Blind Based (we can use Time sensitive request)

In the description below, we provide more type along with these three (3) main type of SQL Injection.

## **Error-based SQLi**

Error-based SQLi is an in-band 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 SQLi**

Union-based SQLi is an in-band 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.

## **Inferential SQLi (Blind SQLi)**

Inferential SQL Injection, unlike in-band SQLi, may take longer for an attacker to exploit, however, it is just as dangerous as any other form of SQL Injection. In an inferential SQLi 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 attacks”).

Instead, an attacker is able to reconstruct the database structure by sending payloads, observing the web application’s response and the resulting behavior of the database server, The two types of inferential SQL Injection are Blind-boolean-based SQLi and Blind-time-based SQLi.

### **Boolean-based (content-based) Blind SQLi**

Boolean-based SQL Injection is an inferential 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 SQLi

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.

## **Out-of-band SQLi**


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.

<https://medium.com/@hninja049/example-of-a-error-based-sql-injection-dce72530271c>

- In real environment attacker will not be sitting and doing attack from web page they can use automated tools like “Burpsuite”. Attacker can use burpsuit running machine as a proxy server so all request will pass through burpsuit and it will modify that request with 1000 of possible request to compromise the database.
- Another automated tool is “Havij” which can be used in graphic interface and allow to test any website for SQL injection susceptibility.
- [www.netsparker.com](http://www.netsparker.com) is a site where from we can get its demo or full version to test any web page
- To learn more about SQL Injection visit [www.owasp.org](http://www.owasp.org) then search for “testing for SQL Injection”

- Reducing risk of SQL Injection attack
  -  Firewall implementation
  - Web Application Firewall / IPS / IDS
  - Web application correct configuration to check size, type, content in request going to database.

- Here is the SQL injection strings..  
Use them in the password field and see the magic.

' or 1=1--

" or 1=1--

or 1=1--

' or 'a'='a

" or "a"="a

') or ('a'='a

") or ("a"="a

## SOME SQL\_COMMANDS

The one that we mostly like to use is this one —> '1' or  
'1=1'

If a user\_name is “admin”, we will bypass the password field like: ‘1’ or ‘1=1’. We copy and paste it in the password field. NOTE: We copy only the underlined ones, without those external quotes. Likewise, if we intend to ignore the password field, we can just type in the username field: ‘admin’ or ‘1=1’, again without the two external quotes, and we leave the password field empty.

- Some commands in SQL

`;(semicolon)` – It means it is end of SQL statement

`‘(Single Quote)` – To represent the close of the SQL statement

`--(double dash)` – It is a comment does not go in process.

`/*comments*/` - It is also a way of adding a comment

- So for example at user logging page if attacker type

Username:-

It will check in database for student user or any user.

-- means anything after that will not be checked