SQL INJECTION (SQLI)

SQL Injection (SQLi) is a type of an injection attack that makes it possible to execute malicious SQL statements. These statements control a database server behind a web application. Attackers can use SQL Injection vulnerabilities to bypass application security measures. An SQL Injection vulnerability may affect any website or web application that uses an SQL database such as MySQL, Oracle, SQL Server, or others. Criminals may use it to gain unauthorized access to your sensitive data: customer information, personal data, trade secrets, intellectual property, and more. SQL Injection attacks are one of the oldest, most prevalent, and most dangerous web application vulnerabilities. The OWASP organization (Open Web Application Security Project) lists injections in their OWASP Top 10 2017 document as the number one threat to web application security

SQL Injection Attack Performed SQL is a query language that was designed to manage data stored in relational databases. You can use it to access, modify, and delete data. Many web applications and websites store all the data in SQL databases. Successful SQL Injection attack can have very serious consequences.

❖ Attackers can use SQL Injections to find the credentials of other users in the database.

❖ An SQL Injection vulnerability could allow the attacker to gain complete access to all data in a database server.

❖ An attacker could use SQL Injection to alter balances, void transactions, or transfer money to their account.

❖ Attacker can delete records from a database or even drop tables.

❖ An attacker could use an SQL Injection as the initial vector and then attack the internal network behind a firewall.

SQL Injection can be classified into three major categories –

1. In-band SQLi,

2. Inferential SQLi and

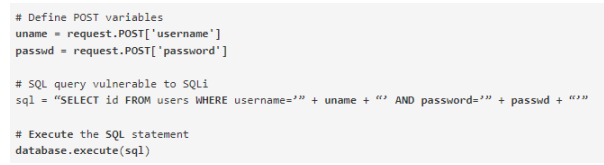
3. Out-of-band SQLi.

1. In-band SQLi (Classic SQLi) 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

i. Error-based SQLi and

ii. Union-based SQLi. 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. 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

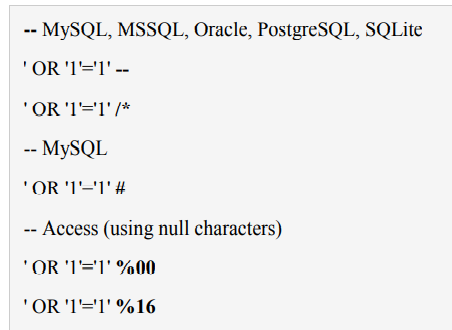
The following script is a simple example of authenticating with a username and a password. The example database has a table named users with the following columns: username and password



These input fields are vulnerable to SQL Injection. An attacker could use SQL commands in the input in a way that would alter the SQL statement executed by the database server. For example, they could use a trick involving a single quote and set the passwd field to: password' OR 1=1 As a result, the database server runs the following SQL query:



Because of the OR 1=1 statement, the WHERE clause returns the first id from the users table no matter what the username and password are. The first user id in a database is very often the administrator. In this way, the attacker not only bypasses authentication but also gains administrator privileges. They can also comment out the rest of the SQL statement to control the execution of the SQL query further



Union-Based SQL Injection One of the most common types of SQL Injection uses the UNION operator. It allows the attacker to combine the results of two or more

SELECT statements into a single result. The technique is called union based SQL Injection. The following is an example of this technique. It uses the web page testphp.vulnweb.com, an intentionally vulnerable website hosted by Acunetix. The following HTTP request is a normal request that a legitimate user would send



The artist parameter is vulnerable to SQL Injection. The following payload modifies the query to look for an inexistent record. It sets the value in the URL query string to -1. Of course, it could be any other value that does not exist in the database. However, a negative value is a good guess because an identifier in a database is rarely a negative number. In SQL Injection, the UNION operator is commonly used to attach a malicious SQL query to the original query intended to be run by the web application. The result of the injected query will be joined with the result of the original query. This allows the attacker to obtain column values from other tables

SQL INJECTION FOR WEBSITE HACKING

**Step 1: Finding Vulnerable Website:**

We can find the Vulnerable websites(hackable websites) using Google Dork list. google dork is searching for vulnerable websites using the google searching tricks. But we are going to use “inurl:” command for finding the vulnerable websites. Some Examples: inurl:index.php?id= inurl:gallery.php?id=

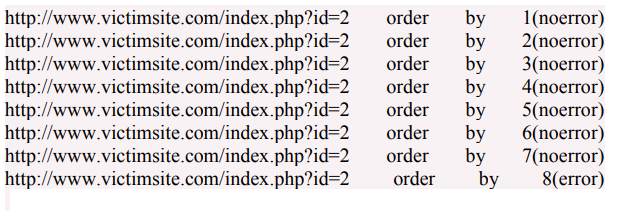
inurl:article.php?id= inurl:pageid=

Here is the huge list of Google Dork <http://www.ziddu.com/download/13161874/A…t.zip.html>

So Start from the first website

**Step 2: Checking the Vulnerability:** In order to check the vulnerability ,add the single quotes(‘) at the end of the url and hit enter. For eg: http://www.victimsite.com/index.php?id=2' If the page remains in same page or showing that page not found or showing some other webpages. Then it is not vulnerable. If it showing any errors which is related to sql query,then it is vulnerable. Cheers..!! For eg: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near ”’ at line 1

**Step 3: Finding Number of columns:** Now we have found the website is vulnerable. Next step is to find the number of columns in the table. For that replace the single quotes(‘) with “order by n” statement.(leave one space between number and order by n statement) Change the n from 1,2,3,4,,5,6,…n. Until you get the error like “unknown column “. change the number until you get the error as “unknown column” if you get the error while trying the “x”th number, then no of column is “x-1”.



so now x=8 , The number of column is x-1 i.e, 7. Sometime the above may not work. At the time add the “–” at the end of the statement

**Step 4: Displaying the Vulnerable columns:** Using “union select columns\_sequence” we can find the vulnerable part of the table. Replace the “order by n” with this statement. And change the id value to negative(i mean id=-2,must change,but in some website may work without changing). Replace the columns\_sequence with the no from 1 to x-1(number of columns) separated with commas(,). It will show some numbers in the page(it must be less than ‘x’ value, i mean less than or equal to number of columns). Like this

Now select 1 number. It showing 3,7. Let’s take the Number 3.

**Step 5: Finding version,database,user** Now replace the 3 from the query with “version()” It will show the version as 5.0.1 or 4.3. something like this. Replace the version() with database() and user() for finding the database,user respectively.

**Step 6: Finding the Table Name if the version is 5 or above**. Then follow these steps. Now we have to find the table name of the database. Replace the 3 with “group\_concat(table\_name) and add the “from information\_schema.tables where table\_schema=database()” Now it will show the list of table names. Find the table name which is related with the admin or user.

Now select the “admin ” table. if the version is 4 or some others, you have to guess the table names. (user, tbluser). It is hard and bore to do sql inection with version 4.

**Step 7: Finding the Column Name** Now replace the “group\_concat(table\_name) with the “group\_concat(column\_name)” Replace the “from information\_schema.tables where table\_schema=database()–” with “FROM information\_schema.columns WHERE table\_name=mysqlchar– Now listen carefully ,we have to find convert the table name to MySql CHAR() string and replace mysqlchar with that . Find MysqlChar() for Tablename: First of all install the HackBar addon: https://addons.mozilla.org/en-US/firefox/addon/3899/ Now select sql->Mysql->MysqlChar()