So far, we have learnt 3 types of SQL Injections:

1. Basic Authentication Bypass

2. GET based SQL Injections

3. POST based SQL Injections

Practicing these SQL Injection methods are good to get you started. But there are other SQL Injection methods that you should know of. So we will briefly explain 3 more SQL Injection methods which you may encounter while researching or practicing.

Error Based SQL Injections:

Sometimes, we cannot exploit SQL Injection vulnerabilities simply by using UNION command. This may be because of some security checks in place or because of the complexity of the code. So to perform error based SQL injections, we make websites to throw SQL errors through which we can extract critical information. Now, different database servers employs different approach of performing error based SQL injections as the errors they throw are different in nature.

For better understanding, let us have a look at the example below:

In Microsoft SQL server, there is an SQL function called convert(), which is used to convert the second parameter to the data type given in the first parameter.

Have a look at the syntax: convert(<data type>,<value>)

This means, if we use convert(int,’145’), the output will be 145.

But, what if we try to convert a value which is not a valid data type like this convert(int,’abcd’)

As you might have expected, the server will throw an error saying:

“Cannot convert string ‘abcd’ into an int”

So, our motive is to perform SQL injection. This means, instead of using convert(int,’abcd) we ask the SQL server to convert(int,db\_name()). As you know, db\_name() is same as database() and suppose the database name is ‘secret\_database’. If we try to convert it, the server will throw an error saying:

“Cannot convert ‘secret\_database’ into int”

Now, if a website throws a message that shows SQL errors, this means we can definitely perform SQL injection here. Using SQL injection, we can easily retrieve the name of the database. And once the database name is known we can easily fetch the names of the tables, columns and finally the data too. These SQL injections are referred to as Error based SQL injection, where we perform SQL injections when a web application throws an SQL error.

Boolean Based Blind Injections:

To understand the injection, let’s fragment it as Boolean + Blind Injections.

So, Boolean in terms of programming simply means True or False. This means, while performing these injections, we might be asking server to respond us as either true or false.

Now, the second part is Blind Injections or Blind SQL injections. As the name suggests, these injections are used where we are successfully able to fetch critical data but somehow the extracted data is not visible on the website (hence, the name blind), which may be attributed to how the website is build.

So, combining both these parts, in Boolean based blind injections, we perform SQL injections by asking server True or False questions and on the basis of the response, we can extract crucial information.

Let’s have a look at the example below:

Suppose, If we want to fetch name of a student from a website, we will simply use this SQL query

Select name from students where id=121

The output will be the name of the student against the id 121.

Now, to perform Boolean based blind injections, we use AND operator. Have a look at the query below where we have used boolean based blind injection to fetch the name of the student.

Select name from students where id=121 AND 1=1+

As 1 equals to 1 us universally true, the output will fetch the name of the student. So, how is this injection different from others as we are just extracting the same information in a different way.

Well, what if we use this query instead.

Select name from students where id=121 AND 1=0+

Now, 1 can never be equal to zero, this means the output will be blank. So, in such cases boolean based blind injections comes into play. This is how the query will look like:

Select name from students where id=121 AND (get\_first\_character\_of(password))=’a’--+

Look carefully, this time we are asking server to tell us the first character as ‘true’ or ‘false’. If the output shows the student name, this means the password starts from ‘a’ and we can proceed further in a similar way to fetch the complete password and if there is no output, it means that the password must start with some other letter. This is how, Boolean based Blind Injections are performed.

Time Based Blind Injections:

These injections are used in those cases where we fail to extract data either by using UNION or ERROR based SQL injections and can neither ask a website questions as True or False. So, in order to extract critical information, we tamper with the server response time.

Whenever a request is made to the server, it takes some time to fetch the information and deliver it to us, this is called as response time. Now, if we tamper with this response time, we can extract some crucial information.

The syntax of Time based Blind injections is similar to Boolean based blind injections. Have a look at the query for time based blind injections.

Select name from students where id=121 AND (if the 1st character of the password = ‘a’ then sleep for 10 seconds)--+

Here, you can see, we are asking the server to tell us the first character of the password. If the password starts from ‘a’, the server will sleep for 10 seconds, which means increase in response time by 10 seconds. And, if the password does not start with ‘a’, the server will take it’s usual response time. In a similar way, we can predict the whole password. This is how, Time based blind injections are performed.

Using this injection has lot of disadvantages. Firstly, as you can see every time we are making a request to server, it sleeps for 10 seconds. This means, this injection will take a lot of time. Secondly, the response time is also dependent on the speed of the internet. If the connection drops in between, it will increase the response time and hence will lead to faulty results.