

## Input Validation Issues- Part 1

Aakash R  
CB.SC.P2CYS23011

We explore the application by entering values in the search EditText field

### 7. Input Validation Issues - Part 1

**Objective:** Try to access all user data without knowing any user name. There are three users by default and your task is to output data of all the three users with a single malicious search.

**Hint:** Improper or no input validation issue arise when the input is not filtered or validated before using it. When developing components that take input from outside, always validate it. For ease of testing there are three users already present in the database, for example one of them is admin, you can try searching for admin to test the output.

User: (admin) pass: (passwd123) Credit card:  
(1234567812345678)

admin

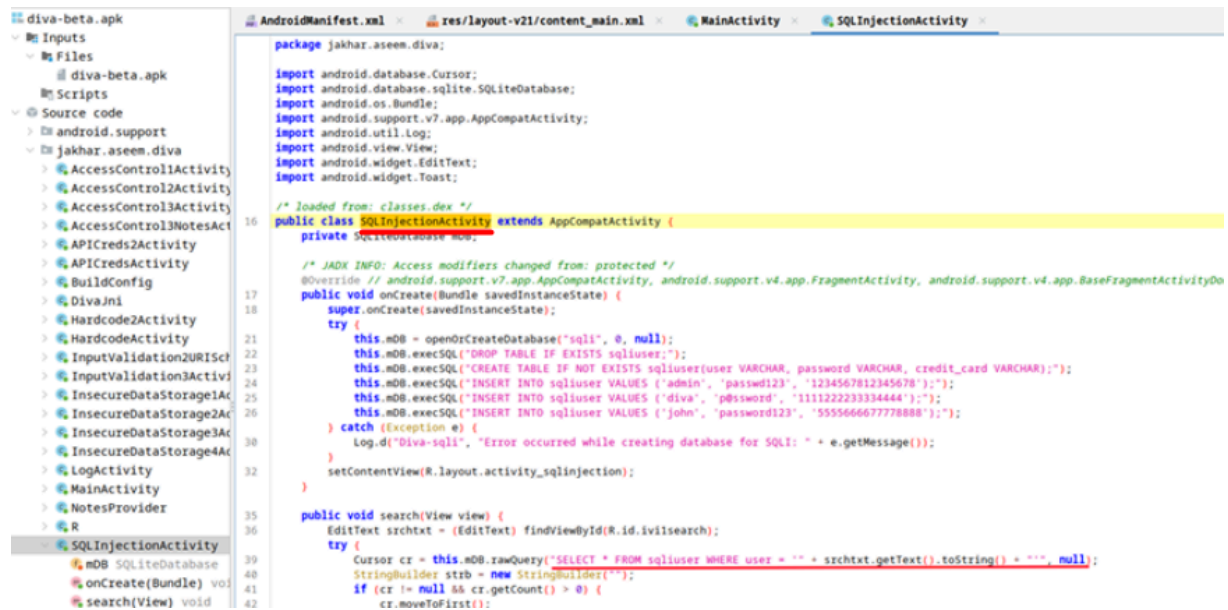
SEARCH

We observe the decompiled source code in the JADX.

```
diva-beta.apk
├── Inputs
│   └── Files
│       └── diva-beta.apk
│           └── Scripts
├── Source code
│   ├── android.support
│   └── jakhar.aseem.diva
│       ├── AccessControl1Activity
│       ├── AccessControl2Activity
│       ├── AccessControl3Activity
│       ├── AccessControl3NotesAct
│       ├── APICreds2Activity
│       ├── APICredsActivity
│       ├── BuildConfig
│       ├── DivaJni
│       ├── Hardcode2Activity
│       ├── HardcodeActivity
│       ├── InputValidation2URISch
│       ├── InputValidation3Activi
│       ├── InsecureDataStorage1Ac
│       ├── InsecureDataStorage2Ac
│       ├── InsecureDataStorage3Ac
│       ├── InsecureDataStorage4Ac
│       ├── LogActivity
│       └── MainActivity
└── AndroidManifest.xml
└── res/layout-v21/content_main.xml
└── MainActivity

57      Intent i3 = new Intent(this, (Class<?>) InsecureDataStorage1Activity.class);
58      startActivity(i3);
100     return;
}
60     if (view == findViewById(R.id.d4button)) {
61         Intent i4 = new Intent(this, (Class<?>) InsecureDataStorage2Activity.class);
62         startActivity(i4);
100     return;
}
64     if (view == findViewById(R.id.d5button)) {
65         Intent i5 = new Intent(this, (Class<?>) InsecureDataStorage3Activity.class);
66         startActivity(i5);
100     return;
}
68     if (view == findViewById(R.id.d6button)) {
69         Intent i6 = new Intent(this, (Class<?>) InsecureDataStorage4Activity.class);
70         startActivity(i6);
100     return;
}
72     if (view == findViewById(R.id.d7button)) {
73         Intent i7 = new Intent(this, (Class<?>) SQLInjectionActivity.class);
74         startActivity(i7);
100     return;
}
76     if (view == findViewById(R.id.d8button)) {
77         Intent i8 = new Intent(this, (Class<?>) InputValidation2URISchemeActivity.class);
78         startActivity(i8);
100     return;
}
80     if (view == findViewById(R.id.d9button)) {
81         Intent i9 = new Intent(this, (Class<?>) AccessControl1Activity.class);
82         startActivity(i9);
}
```

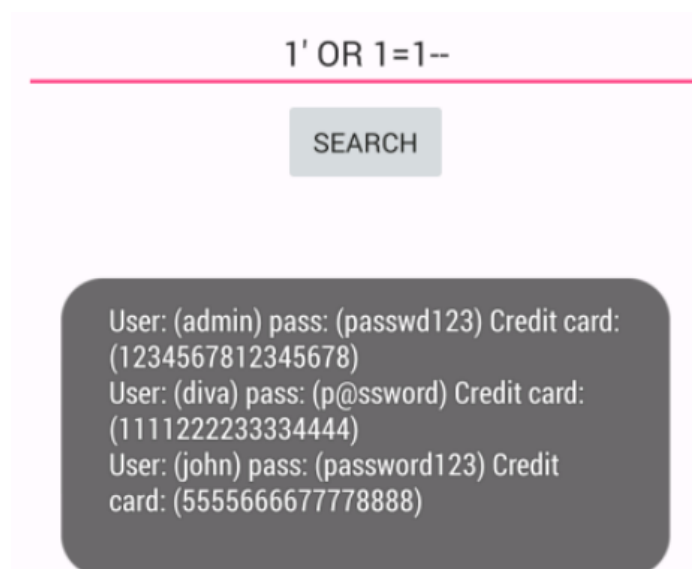
We understand that for this Task an activity called **SQLInjectionActivity** is used.



We enter 1' OR 1=1-- so that the above SQL query becomes

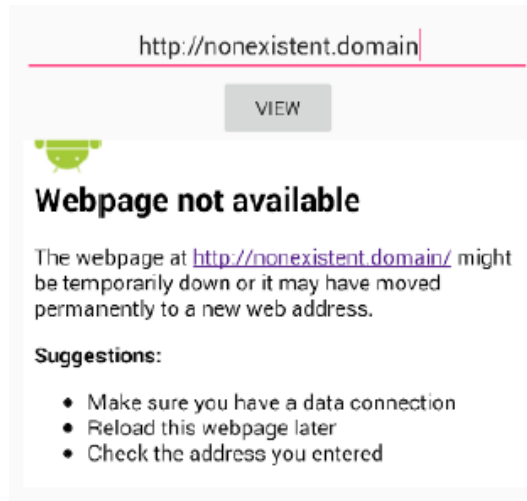
***SELECT \* FROM sqluser WHERE user = '1' OR 1=1--'***

The WHERE clause condition gets evaluated to FALSE or TRUE which is equivalent to TRUE, hence all the records in the database are displayed in the Toast message.



## Input Validation Issues- Part 2

We explore the application by entering values in the URL EditText field.

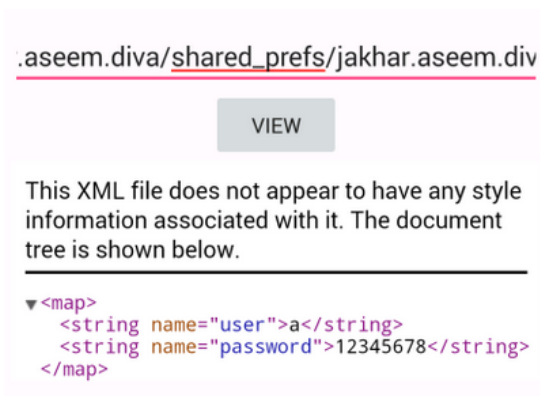


We enter a sensitive path like

***file:///data/data/jakhar.aseem.diva/shared\_prefs/jakhar.aseem.diva\_preferences.xml***

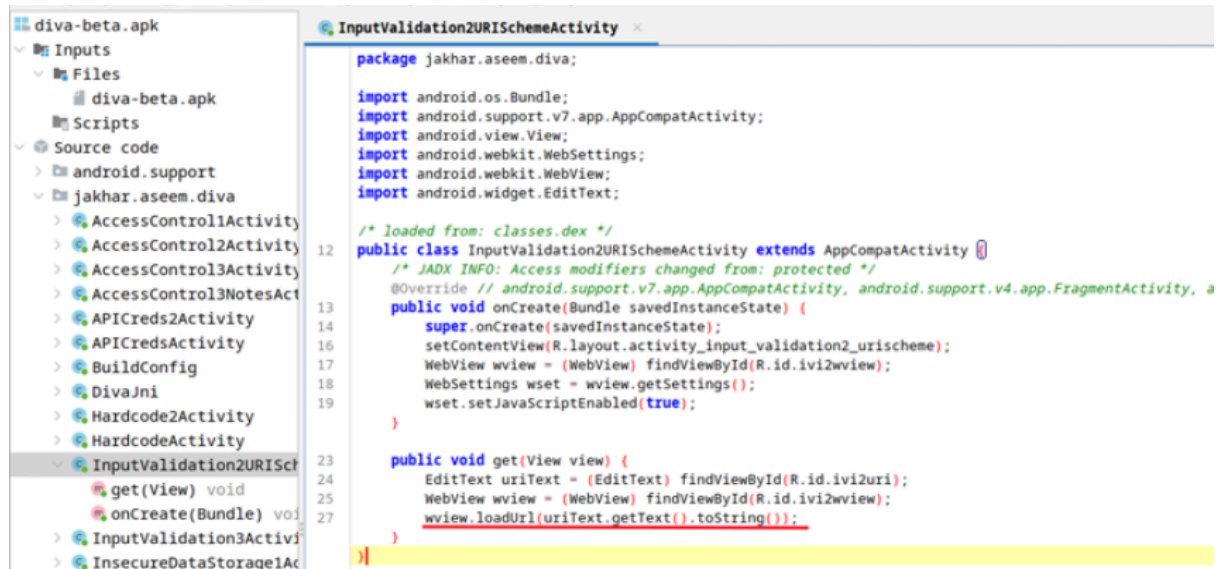
Which only the application has access to and normal user of the device does not have access to. We observe that the file contents are displayed in the WebView.

```
root@generic_x86:/ # su
data/data/jakhar.aseem.diva/
root@generic_x86:/data/data/jakhar.aseem.diva # ls
app_webview
cache
databases
lib
root@generic_x86:/data/data/jakhar.aseem.diva # id
uid=0(root) gid=0(root)
root@generic_x86:/data/data/jakhar.aseem.diva # pwd
/data/data/jakhar.aseem.diva
root@generic_x86:/data/data/jakhar.aseem.diva # ls
app_webview
cache
databases
lib
root@generic_x86:/data/data/jakhar.aseem.diva # ls
app_webview
cache
databases
```



We observe the decompiled source code and open the InputValidation2URISchemeActivity in the JADX.

We observe that the user input value in the EditText field is used directly to load in the WebView without any sanitization or validation.



```
package jakhar.aseem.diva;

import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.webkit.WebSettings;
import android.webkit.WebView;
import android.widget.EditText;

/* loaded from: classes.dex */
public class InputValidation2URISchemeActivity extends AppCompatActivity {
    /* JADX INFO: Access modifiers changed from: protected */
    @Override // android.support.v7.app.AppCompatActivity, android.support.v4.app.FragmentActivity, android.support.design.widget.TextInputLayout
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_input_validation2_urischeme);
        WebView wview = (WebView) findViewById(R.id.iv12wview);
        WebSettings wset = wview.getSettings();
        wset.setJavaScriptEnabled(true);
    }

    public void get(View view) {
        EditText uriText = (EditText) findViewById(R.id.iv12uri);
        WebView wview = (WebView) findViewById(R.id.iv12wview);
        wview.loadUrl(uriText.getText().toString());
    }
}
```