OWASP Top 10 Vulnerabilities in Mobile

Aakash R
CB.SC.P2CYS23011

We will use Diva_beta.apk (Damn Insecure Vulnerable Application) with list of Top 10 Mobile Vulnerabilities.

We install the APK to the emulator by typing the below command.

adb install diva-beta.apk

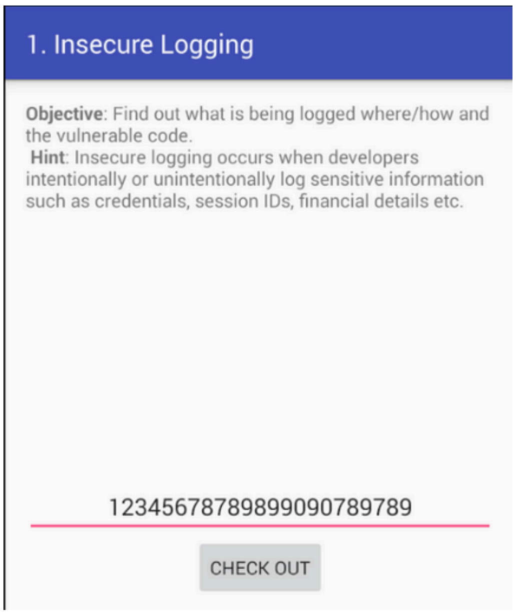
1. Insecure Logging

We type ***adb logcat*** in the terminal window.

adb logcat | grep -i "credit card" or adb logcat | Select-String "cred"

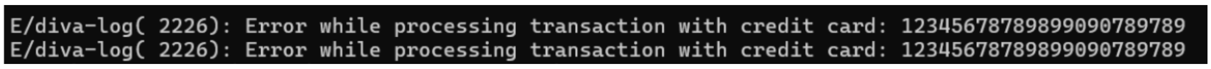
We click on the Login Button

We type a number in the EditText field and click on the Check Out button. We observe that ***An error occurred toast message is shown, and that the logcat has logged the input that was entered.***


1. Insecure Logging

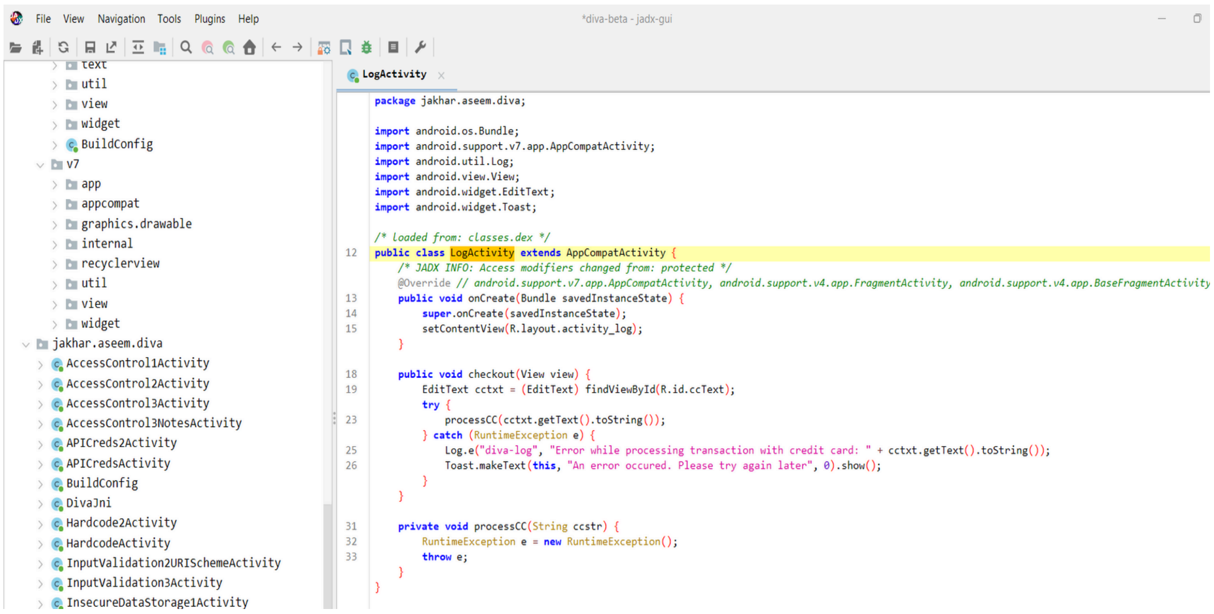
Objective: Find out what is being logged where/how and the vulnerable code.
Hint: Insecure logging occurs when developers intentionally or unintentionally log sensitive information such as credentials, session IDs, financial details etc.

12345678789899090789789


```
E/diva-log( 2226): Error while processing transaction with credit card: 12345678789899090789789
E/diva-log( 2226): Error while processing transaction with credit card: 12345678789899090789789
```

We open JADX and open the **diva-beta.apk** file.

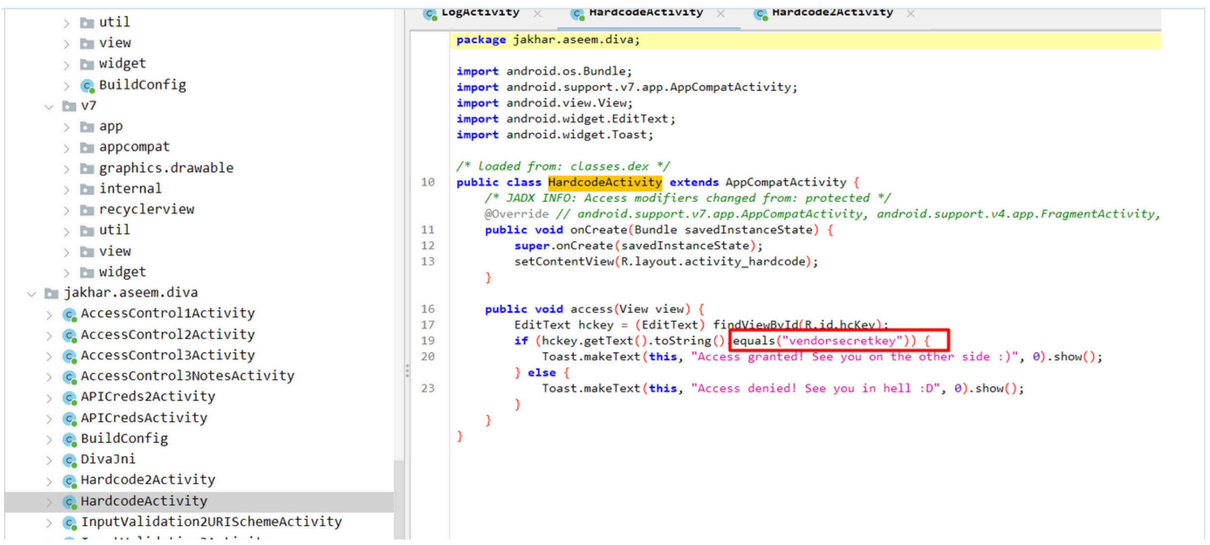
We observe the decompiled source code and open the LogActivity in the JADX



```
package jakhar.aseem.diva;

import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;

/* loaded from: classes.dex */
public class LogActivity extends AppCompatActivity {
    /* JADX INFO: Access modifiers changed from: protected */
    @Override // android.support.v7.app.AppCompatActivity, android.support.v4.app.FragmentActivity, android.support.v4.app.BaseFragmentActivity
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_log);
    }

    public void checkout(View view) {
        EditText cctxt = (EditText) findViewById(R.id.ccText);
        try {
            processCC(cctxt.getText().toString());
        } catch (RuntimeException e) {
            Log.e("diva-log", "Error while processing transaction with credit card: " + cctxt.getText().toString());
            Toast.makeText(this, "An error occured. Please try again later", 0).show();
        }
    }

    private void processCC(String ccstr) {
        RuntimeException e = new RuntimeException();
        throw e;
    }
}
```

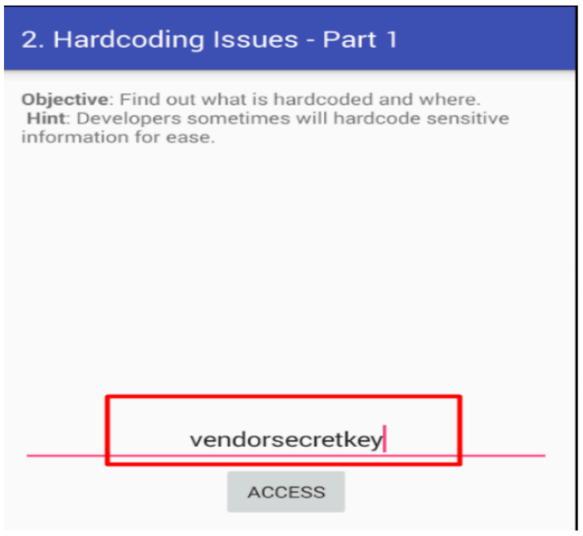
2. Hardcoding Issues - Part 1

When we decompile the source code in Jadx app then it will show the code which will also show some hardcoded string.



```
package jakhar.aseem.diva;

import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;

/* loaded from: classes.dex */
public class HardcodeActivity extends AppCompatActivity {
    /* JADX INFO: Access modifiers changed from: protected */
    @Override // android.support.v7.app.AppCompatActivity, android.support.v4.app.FragmentActivity,
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_hardcode);
    }

    public void access(View view) {
        EditText hckey = (EditText) findViewById(R.id.hcKey);
        if (hckey.getText().toString().equals("vendorsecretkey")) {
            Toast.makeText(this, "Access granted! See you on the other side :)", 0).show();
        } else {
            Toast.makeText(this, "Access denied! See you in hell :D", 0).show();
        }
    }
}
```

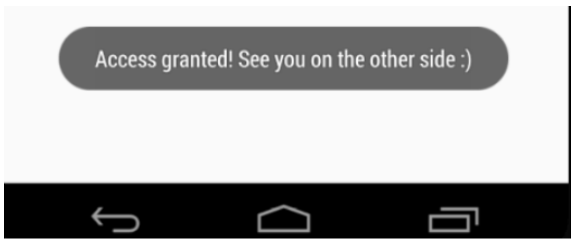
We type the hardcoded vendor key in the EditText field.



2. Hardcoding Issues - Part 1

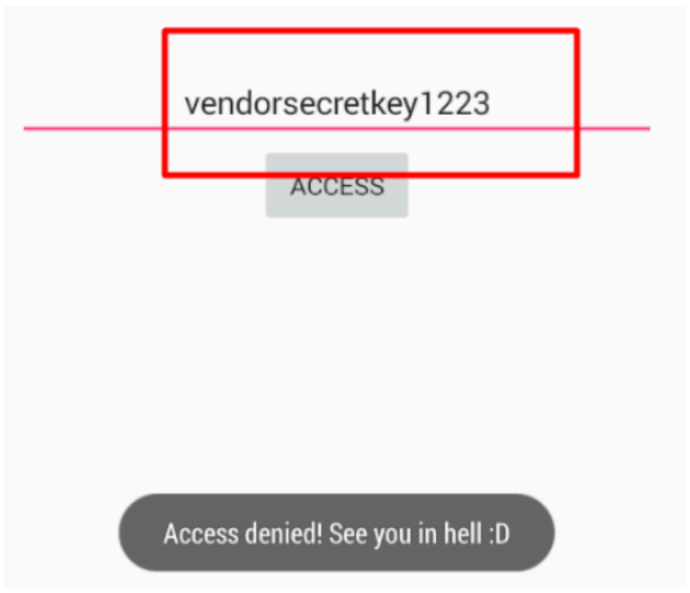
Objective: Find out what is hardcoded and where.
Hint: Developers sometimes will hardcode sensitive information for ease.

vendorsecretkey



Access granted! See you on the other side :)

If we type some other hardcoded key then it will deny.

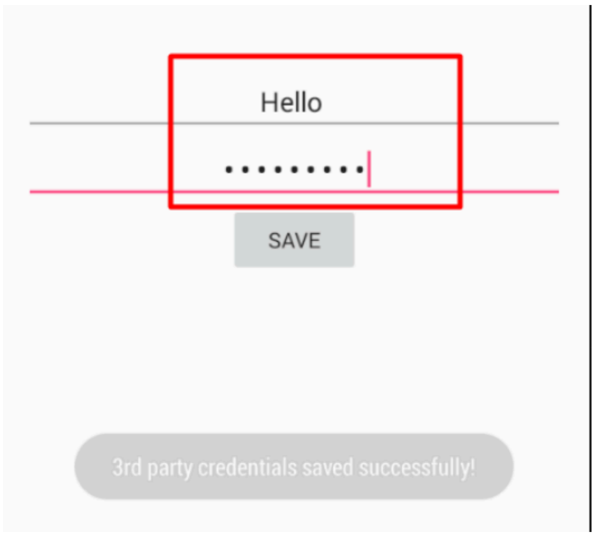


vendorsecretkey1223

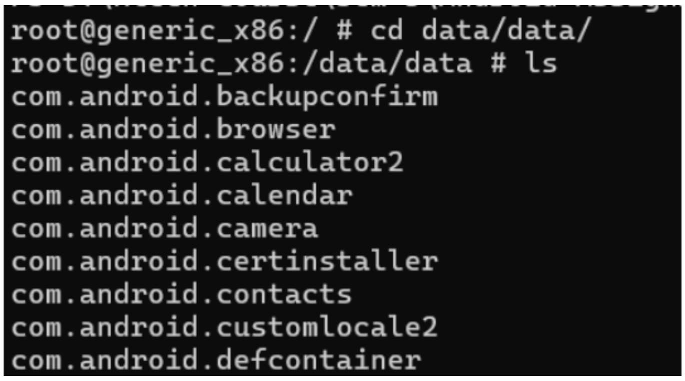
Access denied! See you in hell :D

3. Insecure Data Storage - Part 1

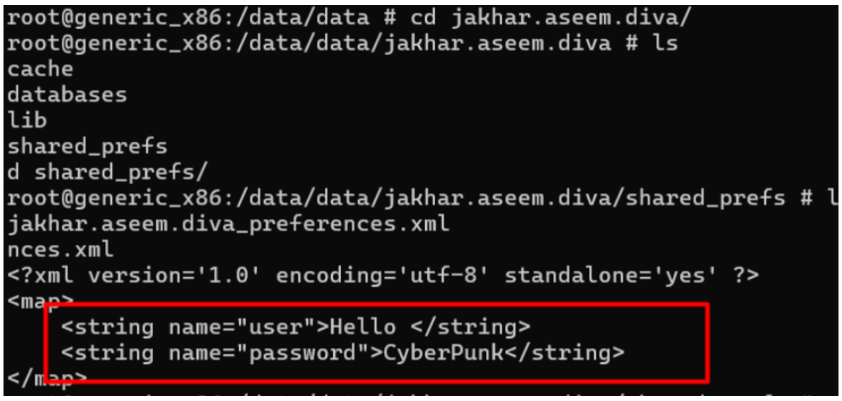
We explore the application by entering username and password in the EditText field.



- We use adb shell to explore the file system used by the application.
- Inside the /data/data/jakhar.aseem.diva directory, we notice the databases and shared_prefs directory.
- We type cat shared_prefs/jakhar.aseem.diva_preferences.xml to see the username and password that were saved by the application.

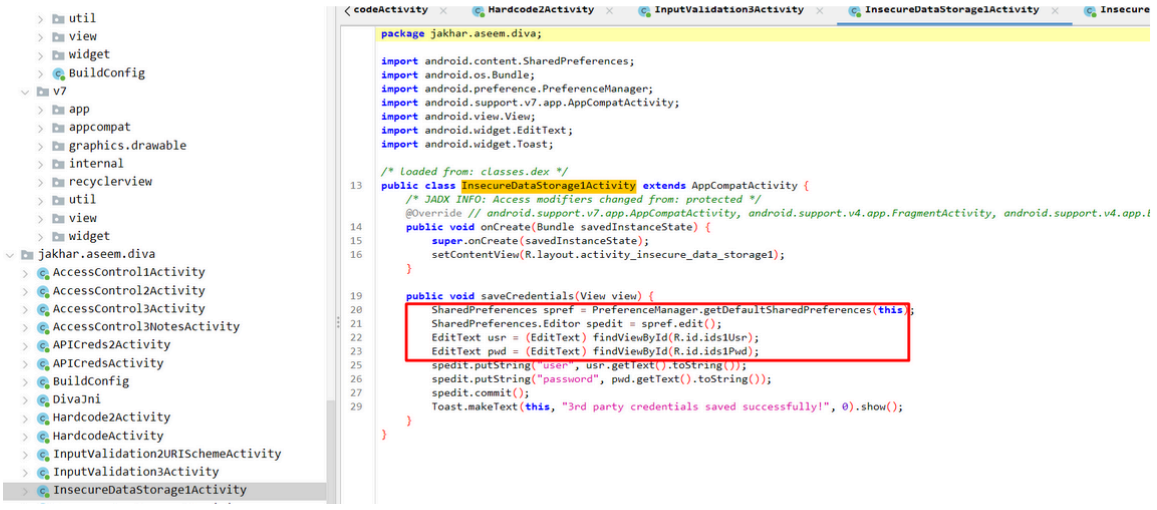


```
root@generic_x86:/ # cd data/data/
root@generic_x86:/data/data # ls
com.android.backupconfirm
com.android.browser
com.android.calculator2
com.android.calendar
com.android.camera
com.android.certinstaller
com.android.contacts
com.android.customlocale2
com.android.defcontainer
```



```
root@generic_x86:/data/data # cd jakhar.aseem.diva/
root@generic_x86:/data/data/jakhar.aseem.diva # ls
cache
databases
lib
shared_prefs
d shared_prefs/
root@generic_x86:/data/data/jakhar.aseem.diva/shared_prefs # l
jakhar.aseem.diva_preferences.xml
nces.xml
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
    <string name="user">Hello </string>
    <string name="password">CyberPunk</string>
</map>
```

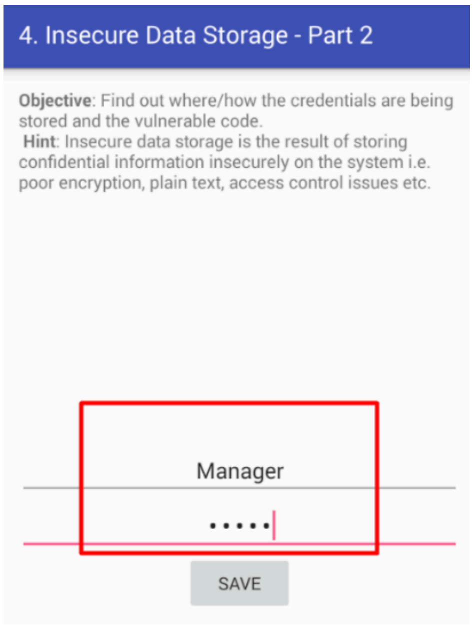
We can observe the decompiled source code and open the InsecureDataStorage1Activity in the Jadx app.



```java
package jakhar.aseem.diva;

import android.content.SharedPreferences;
import android.os.Bundle;
import android.preference.PreferenceManager;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;

/* loaded from: classes.dex */
public class InsecureDataStorage1Activity extends AppCompatActivity {
    /* JADX INFO: Access modifiers changed from: protected */
    @Override // android.support.v7.app.AppCompatActivity, android.support.v4.app.FragmentActivity, android.support.v4.app.
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_insecure_data_storage1);
    }

    public void saveCredentials(View view) {
        SharedPreferences spref = PreferenceManager.getDefaultSharedPreferences(this);
        SharedPreferences.Editor spedit = spref.edit();
        EditText usr = (EditText) findViewById(R.id.ids1Usr);
        EditText pwd = (EditText) findViewById(R.id.ids1Pwd);
        spedit.putString("user", usr.getText().toString());
        spedit.putString("password", pwd.getText().toString());
        spedit.commit();
        Toast.makeText(this, "3rd party credentials saved successfully!", 0).show();
    }
}
```

4. Insecure Data Storage - Part 2

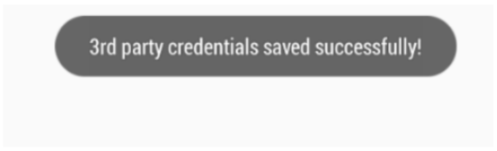
We explore the application by entering username and password in the EditText field.



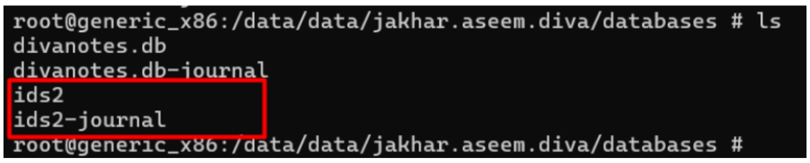
4. Insecure Data Storage - Part 2

Objective: Find out where/how the credentials are being stored and the vulnerable code.
Hint: Insecure data storage is the result of storing confidential information insecurely on the system i.e. poor encryption, plain text, access control issues etc.

Manager
•••••

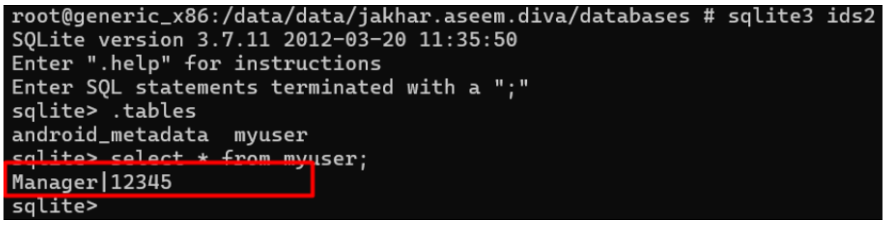


3rd party credentials saved successfully!



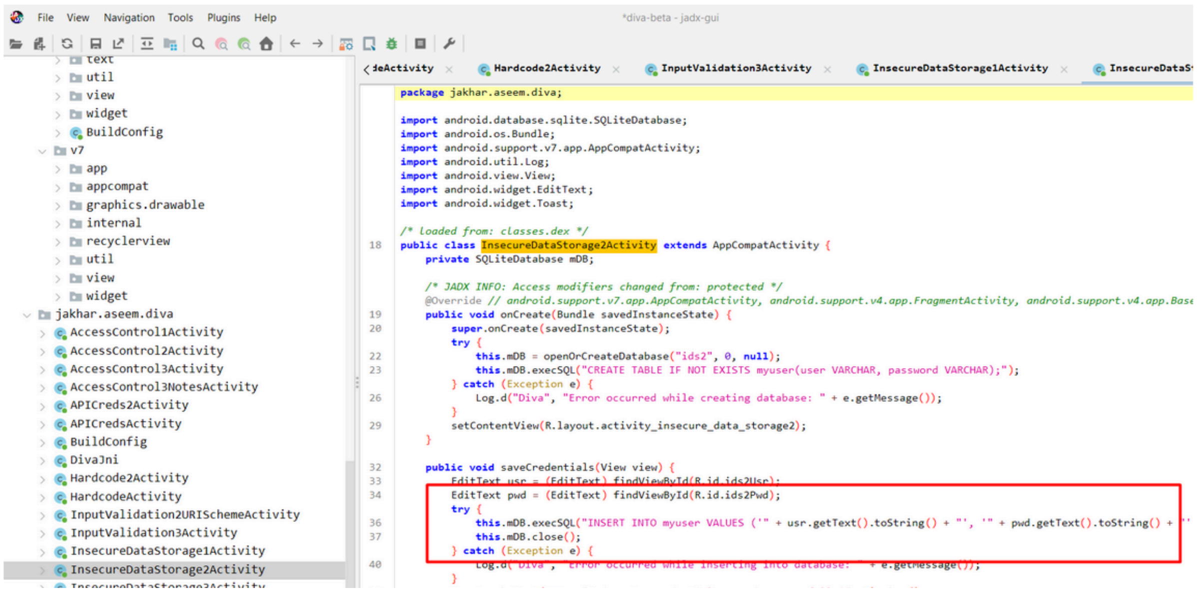
```
root@generic_x86:/data/data/jakhar.aseem.diva/databases # ls
divanotes.db
divanotes.db-journal
ids2
ids2-journal
root@generic_x86:/data/data/jakhar.aseem.diva/databases #
```

We have to open the ids2 database using the sqlite3 program, and enter select * from myuser; to view the saved username and password.



```
root@generic_x86:/data/data/jakhar.aseem.diva/databases # sqlite3 ids2
SQLite version 3.7.11 2012-03-20 11:35:50
Enter ".help" for instructions
Enter SQL statements terminated with a ";"
sqlite> .tables
android_metadata  myuser
sqlite> select * from myuser;
Manager|12345
sqlite>
```

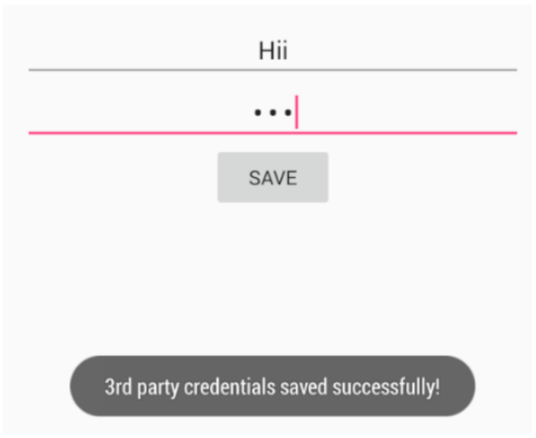
We can view the source code of application by jdax

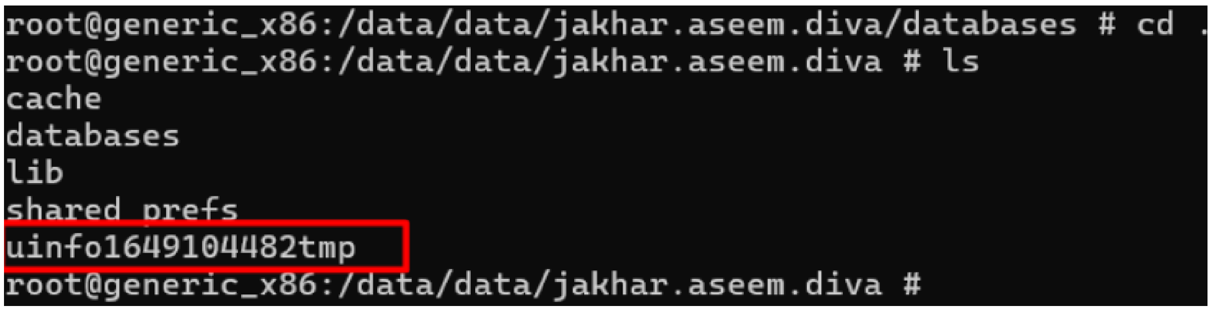


```
package jakhar.aseem.diva;

import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;

/* loaded from: classes.dex */
public class InsecureDataStorage2Activity extends AppCompatActivity {
    private SQLiteDatabase mDB;

    /* JADX INFO: Access modifiers changed from: protected */
    @Override // android.support.v7.app.AppCompatActivity, android.support.v4.app.FragmentActivity, android.support.v4.app.Base
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        try {
            this.mDB = openOrCreateDatabase("ids2", 0, null);
            this.mDB.execSQL("CREATE TABLE IF NOT EXISTS myuser(user VARCHAR, password VARCHAR);");
        } catch (Exception e) {
            Log.d("Diva", "Error occurred while creating database: " + e.getMessage());
        }
        setContentView(R.layout.activity_insecure_data_storage2);
    }

    public void saveCredentials(View view) {
        EditText usr = (EditText) findViewById(R.id.ids2Usr);
        EditText pwd = (EditText) findViewById(R.id.ids2Pwd);
        try {
            this.mDB.execSQL("INSERT INTO myuser VALUES ('" + usr.getText().toString() + "', '" + pwd.getText().toString() + "'
            this.mDB.close();
        } catch (Exception e) {
            Log.d("Diva", "Error occurred while inserting into database: " + e.getMessage());
        }
```

5. Insecure Data Storage - Part 3

We explore the application by entering username and password in the EditText field.

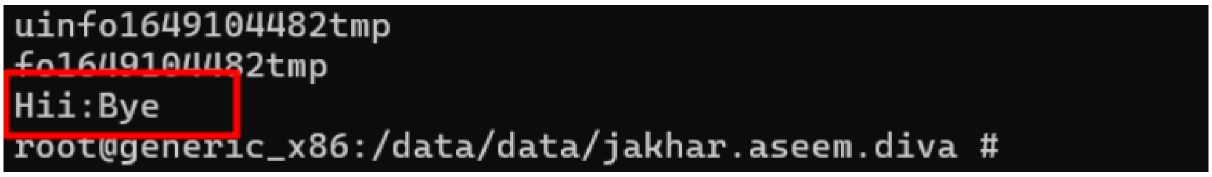




```
root@generic_x86:/data/data/jakhar.aseem.diva/databases # cd .
root@generic_x86:/data/data/jakhar.aseem.diva # ls
cache
databases
lib
shared_prefs
uinfo1649104482tmp
root@generic_x86:/data/data/jakhar.aseem.diva #
```

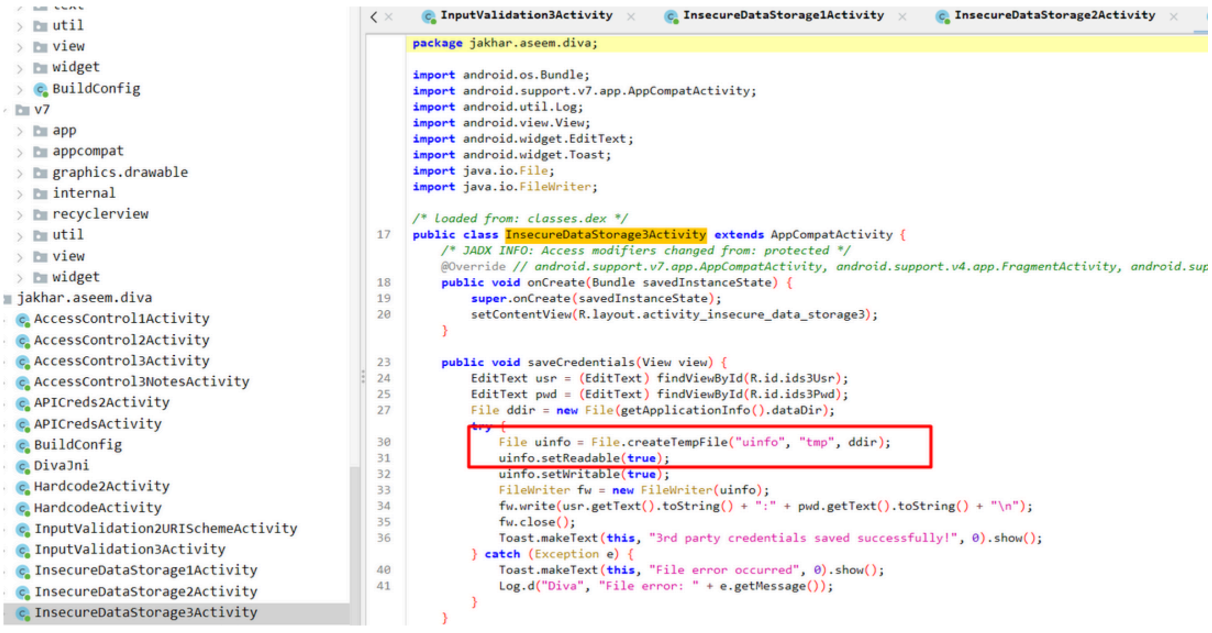
We use adb shell to explore the file system used by the application.

Inside the ***/data/data/jakhar.aseem.diva*** directory, we observe that there is a new file with the name uinfo1649104482tmp. We display the contents of the file using the cat uinfo1649104482tmp command.



```
uinfo1649104482tmp
fo1649104482tmp
Hii:Bye
root@generic_x86:/data/data/jakhar.aseem.diva #
```

We observe the decompiled source code and open the InsecureDataStorage3Activity in the JADX.

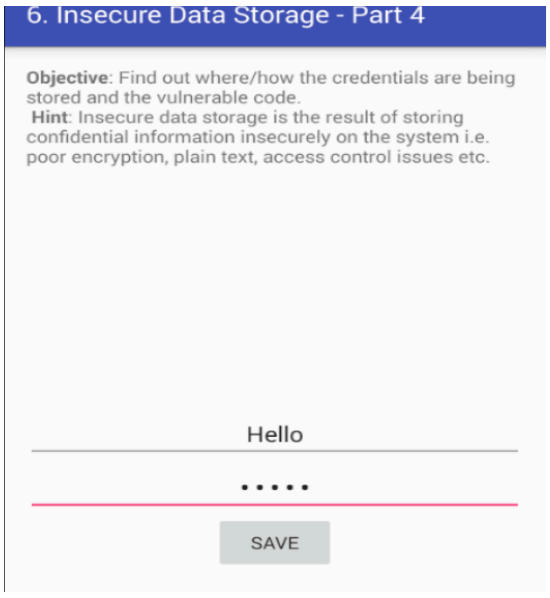


```
> util
> view
> widget
> BuildConfig
v7
> app
> appcompat
> graphics.drawable
> internal
> recyclerview
> util
> view
> widget
jakhar.aseem.diva
AccessControl1Activity
AccessControl2Activity
AccessControl3Activity
AccessControl3NotesActivity
APICreds2Activity
APICredsActivity
BuildConfig
DivaJni
Hardcode2Activity
HardcodeActivity
InputValidation2URISchemeActivity
InputValidation3Activity
InsecureDataStorage1Activity
InsecureDataStorage2Activity
InsecureDataStorage3Activity
```

```java
package jakhar.aseem.diva;

import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;
import java.io.File;
import java.io.FileWriter;

/* loaded from: classes.dex */
public class InsecureDataStorage3Activity extends AppCompatActivity {
    /* JADX INFO: Access modifiers changed from: protected */
    @Override // android.support.v7.app.AppCompatActivity, android.support.v4.app.FragmentActivity, android.sup
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_insecure_data_storage3);
    }

    public void saveCredentials(View view) {
        EditText usr = (EditText) findViewById(R.id.ids3Usr);
        EditText pwd = (EditText) findViewById(R.id.ids3Pwd);
        File ddir = new File(getApplicationInfo().dataDir);
        try {
            File uinfo = File.createTempFile("uinfo", "tmp", ddir);
            uinfo.setReadable(true);
            uinfo.setWritable(true);
            FileWriter fw = new FileWriter(uinfo);
            fw.write(usr.getText().toString() + ":" + pwd.getText().toString() + "\n");
            fw.close();
            Toast.makeText(this, "3rd party credentials saved successfully!", 0).show();
        } catch (Exception e) {
            Toast.makeText(this, "File error occurred", 0).show();
            Log.d("Diva", "File error: " + e.getMessage());
        }
    }
```

Insecure Data Storage - Part 4

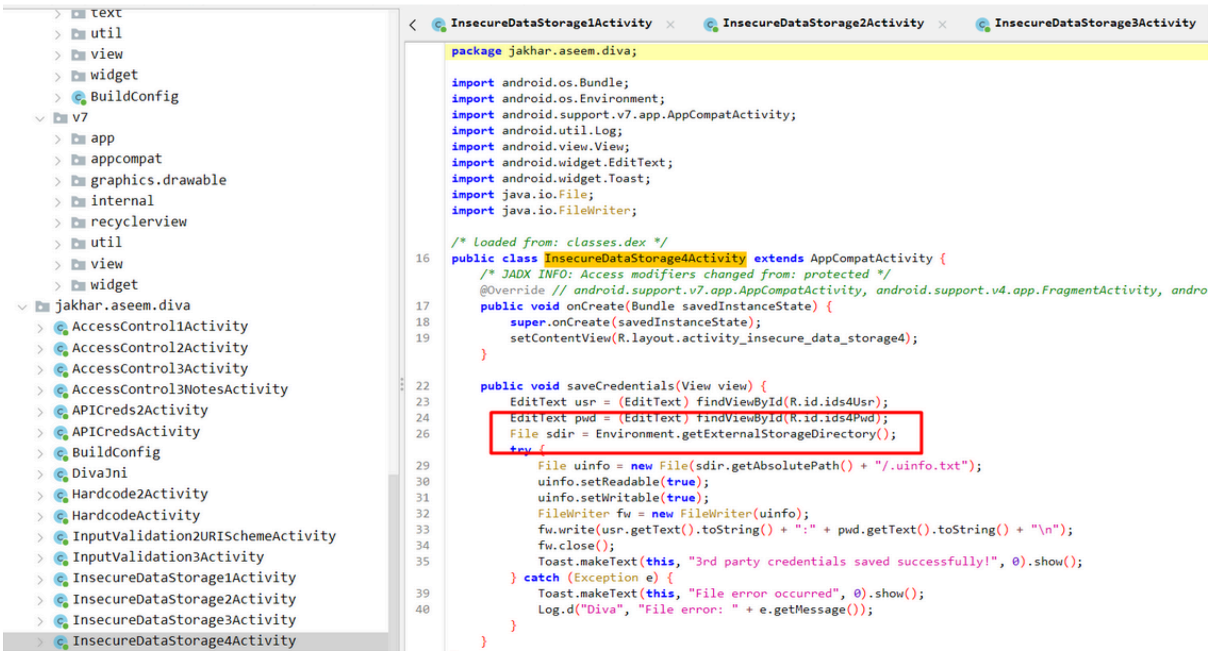
We explore the application by entering username and password in the EditText field.



6. Insecure Data Storage - Part 4

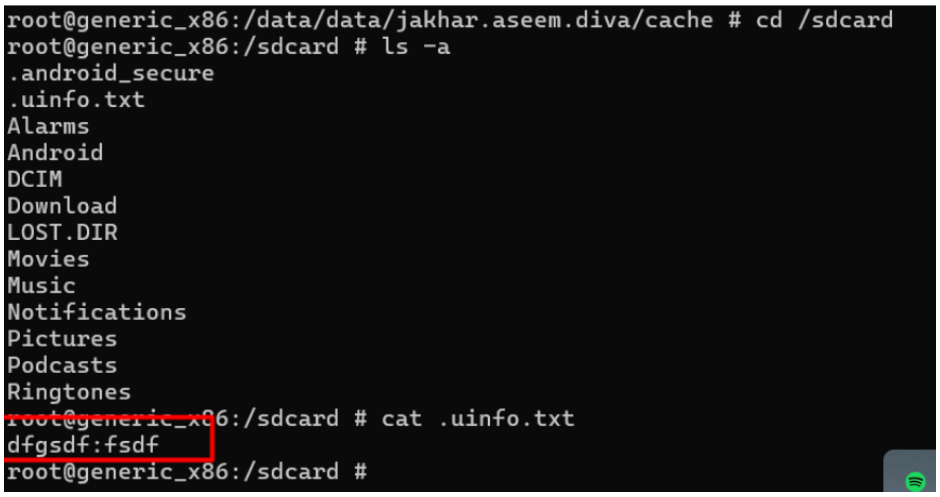
Objective: Find out where/how the credentials are being stored and the vulnerable code.
Hint: Insecure data storage is the result of storing confidential information insecurely on the system i.e. poor encryption, plain text, access control issues etc.

We observe the decompiled source code and open the InsecureDataStorage4Activity in the JADX.



```java
package jakhar.aseem.diva;

import android.os.Bundle;
import android.os.Environment;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;
import java.io.File;
import java.io.FileWriter;

/* loaded from: classes.dex */
public class InsecureDataStorage4Activity extends AppCompatActivity {
    /* JADX INFO: Access modifiers changed from: protected */
    @Override // android.support.v7.app.AppCompatActivity, android.support.v4.app.FragmentActivity, andro
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_insecure_data_storage4);
    }

    public void saveCredentials(View view) {
        EditText usr = (EditText) findViewById(R.id.ids4Usr);
        EditText pwd = (EditText) findViewById(R.id.ids4Pwd);
        File sdir = Environment.getExternalStorageDirectory();
        try {
            File uinfo = new File(sdir.getAbsolutePath() + "/.uinfo.txt");
            uinfo.setReadable(true);
            uinfo.setWritable(true);
            FileWriter fw = new FileWriter(uinfo);
            fw.write(usr.getText().toString() + ":" + pwd.getText().toString() + "\n");
            fw.close();
            Toast.makeText(this, "3rd party credentials saved successfully!", 0).show();
        } catch (Exception e) {
            Toast.makeText(this, "File error occurred", 0).show();
            Log.d("Diva", "File error: " + e.getMessage());
        }
    }
```

As it is located in external storage, In the adb shell, we navigate to cd /sdcard and type the commands: ls -a and cat .uinfo.txt to view the contents of the file.



```
root@generic_x86:/data/data/jakhar.aseem.diva/cache # cd /sdcard
root@generic_x86:/sdcard # ls -a
.android_secure
.uinfo.txt
Alarms
Android
DCIM
Download
LOST.DIR
Movies
Music
Notifications
Pictures
Podcasts
Ringtones
root@generic_x86:/sdcard # cat .uinfo.txt
dfgsdf:fsdf
root@generic_x86:/sdcard #
```